Machine Learning (2022) 111:1049–1072
https://doi.org/10.1007/s10994-021-06071-x

Multiple partitions alignment via spectral rotation

Shudong Huang[1] · Ivor W. Tsang[2] · Zenglin Xu[3] · Jiancheng Lv[1]

Received: 26 May 2021 / Revised: 11 August 2021 / Accepted: 22 September 2021 /
Published online: 6 January 2022

Abstract
Multi-view spectral clustering has drawn much attention due to the effectiveness of exploiting the similarity relationships among data points. These methods typically reveal the intrinsic structure using a predefined graph for each view. The predefined graphs are fused to a consensus one, on which the final clustering results are obtained. However, such common strategies may lead to information loss because of the inconsistency or noise among multiple views. In this paper, we propose to merge multi-view information in partition level instead of the raw feature space where the data points lie. The partition of each view is treated as a perturbation of the consensus clustering, and the multiple partitions are integrated by estimating a distinct rotation for each partition. The proposed model is formulated as a joint learning framework, i.e., with the input data matrix, our model directly outputs the final discrete clustering result. Hence it is an end-to-end single-stage learning model. An iterative updating algorithm is proposed to solve the learning problem, in which the involved variables can be optimized in a mutual reinforcement manner. Experimental results on real-world data sets illustrate the effectiveness of our model.

Keywords Multi-view learning · Clustering · Graph optimization · Spectral rotation

Editors: Yu-Feng Li, Mehmet Gönen, Kee-Eung Kim.

✉ Shudong Huang
huangsd@scu.edu.cn

Ivor W. Tsang
ivor.tsang@uts.edu.au

Zenglin Xu
xuzenglin@hit.edu.cn

Jiancheng Lv
lvjiancheng@scu.edu.cn

[1] College of Computer Science, Sichuan University, Chengdu 610065, China

[2] Centre for Artificial Intelligence, University of Technology Sydney, Sydney NSW 2007, Australia

[3] School of Computer Science and Technology, Harbin Institute of Technology Shenzhen, Shenzhen 518055, China

1 Introduction

Multi-view learning methods aim to exploit the complementary and compatible information among multiple views such that the learning performance can be boosted (Sun 2013; Huang et al. 2020). Among these, multi-view clustering plays a significant part of multi-view learning in unsupervised scenario (Zhang et al. 2019; Lin et al. 2021; Yang et al. 2021).

Due to the efficiency of revealing the similarity relationships between data points, graph-based strategies have been popularly applied in multi-view clustering. Co-regularized multi-view spectral clustering (Kumar et al. 2012), co-training multi-view spectral clustering (Kumar and Daumé 2011), multi-view clustering with kernelized graph learning (Huang et al. 2019), multi-view graph partitioning via discriminative metric learning (Li et al. 2018), multi-view graph embedding for connectome analysis (Ma et al. 2017), weighted multi-view projected clustering with graph learning (Wang et al. 2019), to name a few.

For graph-based multi-view clustering methods, there are mainly three techniques used to construct similarity graph. A dominating one is the classical k-nearest neighbors algorithm, which has been widely utilized to explore the similarity relationships as well as the local manifold structure Hong et al. (2015). For instance, Nie et al. (2017) presented a self-weighted multi-view clustering by introducing a Laplacian rank constrained graph that can be treated as the centroid of the built graph for each view with different confidences. Zong et al. (2018) investigated a weighted multi-view spectral clustering, in which the largest canonical angle is utilized to measure the difference between spectral clustering results. The second type is the so-called self-expressive property, which assumes that each data point can be represented by a linear combination of other points. Gao et al. (2015) proposed a multi-view subspace clustering that enforces each constructed graph to share the same embedded subspace. Huang et al. (2019) introduced a kernelized multi-view subspace clustering by extracting the nonlinear structure in kernel space. The third type concentrated on learning the similarity graph based on adaptive neighbors idea. For example, Feiping Nie and Li (2017) designed a multi-view learning framework with adaptive neighbors, which is able to tackle clustering as well as semi-supervised classification tasks. Wang et al. (2020) constructed an adaptive graph for each view, and then fused the multiple graphs into a unified one to obtain the final clustering results. Yang et al. (2019) established an adaptive sample-level graph fusion mechanism for partial multi-view clustering, where the contributions of distinct views are automatically adjusted. The data samples in this model are partitioned into complete and incomplete parts, while a joint learning strategy is designed to facilitate the similarity between them and thereby improve the final clustering performance.

Despite the progress in multi-view clustering brought by the aforementioned methods, a main drawback is that they mainly merge the heterogeneous information in the raw feature space where the data points lie. Considering the unavoidable noise in data representation, the constructed graphs might be corrupted and cannot reveal the true similarity relationships among data points. For multi-view clustering, the basic principle is that there exists a latent common cluster structure shared by all views. The natural space for multi-view clustering should be all partitions. Hence, partitions from different views might be less affected by noise and easier to achieve an agreement (Nie et al. 2018; Kang et al. 2018). Therefore, it would be more favorable to achieve consensus clustering in partition space.

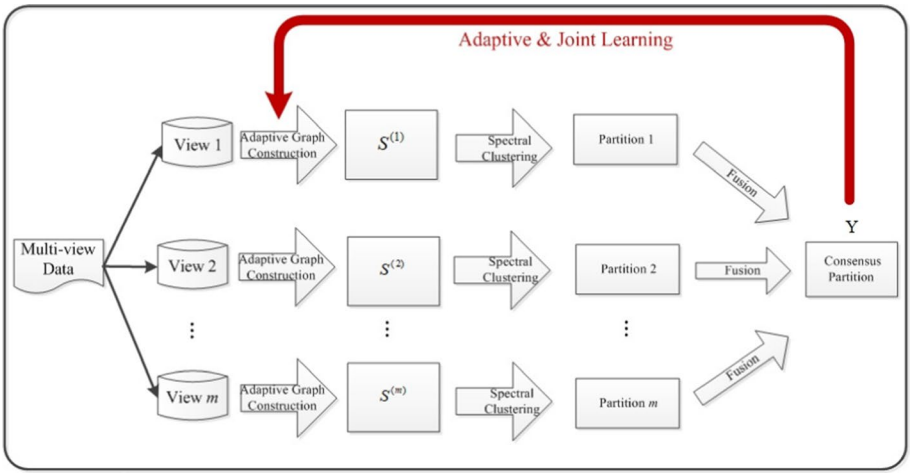

Fig. 1 The framework of our model. The similarity graphs are firstly constructed by utilizing adaptive learning strategy. Then the multiple base partitions are obtained based on the similarity graphs. Finally, the multiple partitions are fused to a consensus partition. These three learning subtasks are mutually boosted until convergence

In this paper, we propose to merge multi-view information in partition level instead of the raw feature space where the data points lie. Inspire by the spectral rotation technique Huang et al. (2013), we consider the partition of each view as a perturbation of the consensus clustering, and the multiple partitions can be integrated by estimating a distinct rotation for each partition. We model the partitions fusion following two intuitive principles: (1) each partition can be essentially seen as a perturbation of the consensus clustering; and (2) the closer a partition to the consensus clustering, the larger weight should be allocated. As a result, different partitions are integrated into a consensus one with reasonable weights. Moreover, considering the predefined graph might not be suitable for the subsequent clustering task. We adopt a joint learning framework that accomplishes from similarity graph construction to final clustering. Hence it is an end-to-end single-stage learning model. An iterative updating algorithm is proposed to solve the learning problem, in which the involved variables can be optimized in a mutual reinforcement manner (Fig. 1).

We outline the salient contributions of our work as follows:

1. *Motivation* This work studies an advanced multi-view graph clustering paradigm and provides a new clustering solution for multi-view data.
2. *Model* We propose a general and effective model to fuse multiple partitions by estimating a distinct rotation for each partition, which is able to merge multi-view information in partition level instead of the raw feature space where the data points lie. Our model accomplishes the subtasks of learning the similarity graphs adaptively, fusing the multiple partitions, and assigning cluster label to each instance in a unified framework. Remarkably, these three learning subtasks can be mutually boosted.
3. *Algorithm* An alternating iterative algorithm with convergence guarantee is proposed to solve the proposed problem, wherein each subproblem can be deduced with an optimal solution.

4. *Results* Extensive experiments on several benchmark multi-view datasets are performed to demonstrate the effectiveness of our model in comparison to the state-of-the-art baseline algorithms.

Notations Throughout this paper, matrices, vectors and scalars are respectively represented by boldface uppercase letters (e.g., $\mathbf{M}$), boldface lowercase letter (e.g., $\mathbf{m}$) and lowercase letter (e.g., m). The ijth element of $\mathbf{M}$ is denoted as m_{ij}. $\mathrm{Tr}(\mathbf{M})$ means the trace of $\mathbf{M}$ and $\|\mathbf{M}\|_F$ is the Frobenius norm of $\mathbf{M}$. $\mathbf{M} \geq 0$ indicates that all elements of $\mathbf{M}$ are nonnegative. $\mathbf{1}$ is a column vector with all its elements being 1. $Ind \overset{\text{def}}{=} \{\mathbf{Y} \in \{0,1\}^{n\times c} | \mathbf{Y1} = \mathbf{1}\}$ indicates a set of indicator matrices.

2 Preliminaries

For dataset $\mathbf{X} = [\mathbf{x}_1, \cdots, \mathbf{x}_n] \in \mathbb{R}^{d\times n}$ with n data samples and d features, the k-nearest neighbor (k-NN) graph is one of the most widely used strategies to construct similarity graph $\mathbf{S} \in \mathbb{R}^{n\times n}$. The ijth element of $\mathbf{S}$, s_{ij}, denotes the similarity between data points $\mathbf{x}_i$ and $\mathbf{x}_j$. It is usually defined by a Gaussian function, $s_{ij} = exp\left(\frac{-\|\mathbf{x}_i - \mathbf{x}_j\|^2}{2\sigma^2}\right)$, where σ determines the width of neighborhoods. One major limitation of k-NN graph is that the hyperparameter σ is difficult to tune because of the negative influence of noise and outliers in data.

Instead of relying on the determinate neighborhood relationship, we intend to learn a probabilistic neighborhood similarity graph. For a data point $\mathbf{x}_i$, other data points $\{\mathbf{x}_j\}_{j=1}^n$ can be treated as the neighborhood of $\mathbf{x}_i$ with probability s_{ij}. Mathematically, we model this thought by solving:

$$\begin{aligned} \min_{\mathbf{S}} \sum_{i,j=1}^{n} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 s_{ij} + \eta \mathcal{R}(\mathbf{S}) \\ \text{s.t.} \quad s_{ii} = 0, s_{ij} \geq 0, \end{aligned} \tag{1}$$

where η is a regularization parameter, and $\mathcal{R}(\mathbf{S})$ denotes a regularization term which is commonly instantiated by sparse learning Huang et al. (2015), low-rank learning Zhang et al. (2013), block diagonal representation Lu et al. (2018), etc. Various studies have shown that sparse representation is robust to noisy data and outliers He et al. (2013). We can simply utilize the sparsity-inducing norm, ℓ_1-norm, to achieve a sparse representation by setting $\mathcal{R}(\mathbf{S}) = \sum_{i=1}^{n} \left\|\mathbf{s}_i\right\|_1$. Thus (1) becomes

$$\begin{aligned} \min_{\mathbf{S}} \sum_{i,j=1}^{n} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 s_{ij} + \eta \sum_{i=1}^{n} \left\|\mathbf{s}_i\right\|_1 \\ \text{s.t.} \quad s_{ii} = 0, s_{ij} \geq 0, \end{aligned} \tag{2}$$

where $\mathbf{s}_i$ is the ith column of $\mathbf{S}$. Typically, we normalize $\mathbf{s}_i^T\mathbf{1} = 1$, thus the regularization term becomes a constant. In other words, the normalization $\mathbf{s}_i^T\mathbf{1} = 1$ actually enforces a sparse constraint on $\mathbf{S}$. Then, (2) can be written as

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{n} \left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 s_{ij} \\ &\text{s.t.} \quad \forall i, \mathbf{s}_i^T \mathbf{1} = 1, s_{ii} = 0, s_{ij} \geq 0. \end{aligned} \tag{3}$$

However, (3) has a trivial solution, i.e., only one data point (the nearest one) can be considered as neighbor of $\mathbf{x}_i$ with probability 1. Following Nie et al. (2014) and Du and Shen (2015), we impose a prior on (3), and thus it can be reformulated as

$$\begin{aligned} &\min_{\mathbf{S}} \sum_{i,j=1}^{n} \left(\left\|\mathbf{x}_i - \mathbf{x}_j\right\|_2^2 s_{ij} + \alpha s_{ij}^2 \right) \\ &\text{s.t.} \quad \forall i, \mathbf{s}_i^T \mathbf{1} = 1, s_{ii} = 0, s_{ij} \geq 0. \end{aligned} \tag{4}$$

where α is a trade-off parameter. The prior $\sum_{i,j=1}^{n} s_{ij}^2$ can be considered as uniform distribution. When $\alpha \to \infty$, we can see all data points are connected to x_i with identical probability $\frac{1}{n}$. Thus it is important to allocate a suitable value to α such that the first term and the second term in (4) can be well balanced, which will be analyzed in a later section.

Once the affinity graph $\mathbf{S}$ is obtained, one can perform the classical spectral clustering algorithm (Ng et al. 2002) on it to get the clustering results:

$$\begin{aligned} &\min_{\mathbf{F}} \operatorname{Tr}\left(\mathbf{F}^T \mathbf{L} \mathbf{F}\right) \\ &\text{s.t.} \quad \mathbf{F} \in \mathbb{R}^{n \times k}, \mathbf{F}^T \mathbf{F} = \mathbf{I}, \end{aligned} \tag{5}$$

where $\mathbf{L} \in \mathbb{R}^{n \times n}$ is the Laplacian of $\mathbf{S}$, $\mathbf{F}$ denotes the spectral embedding and k is the cluster number. Here, $\mathbf{L}$ is given by $\mathbf{L} = \mathbf{D} - \mathbf{S}$, and $\mathbf{D}$ is a diagonal matrix with its i-th diagonal element being $d_{ii} = \sum_j s_{ij}$. In fact, $\mathbf{F}$ can be considered as a continuous partition result. The final discrete clustering result is usually obtained by implementing the standard k-means algorithm on $\mathbf{F}$.

3 The proposed approach

Apparently the aforementioned formulations are designed for single-view data clustering. Despite the promising performance made by these methods, they still cannot deal with the multi-view data. In this paper, we focus on solving the multi-view clustering problem by extending the above formulations such that the multi-view information can be reasonably integrated. Denote $\mathbf{X} = \{\mathbf{X}^{(1)}, \ldots, \mathbf{X}^{(m)} | \mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}\}$ as a multi-view dataset with m views, and d_v is the number of features in the vth view. (4) and (5) can be jointly extended to the multi-view formulation:

$$\begin{aligned} &\min_{\{\mathbf{S}^{(v)}\}, \{\mathbf{F}^{(v)}\}} \sum_{v=1}^{m} \left\{ \sum_{i,j=1}^{n} \left\|\mathbf{x}_i^{(v)} - \mathbf{x}_j^{(v)}\right\|_2^2 s_{ij}^{(v)} + \alpha \left(s_{ij}^{(v)}\right)^2 \right. \\ &\left. + \beta \operatorname{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T \mathbf{L}^{(v)} \mathbf{F}^{(v)} \right) \right\} \\ &\text{s.t.} \quad \forall i, \left(\mathbf{s}_i^{(v)}\right)^T \mathbf{1} = 1, s_{ii}^{(v)} = 0, s_{ij}^{(v)} \geq 0, \mathbf{F}^{(v)} \in \mathbb{R}^{n \times c}, \left(\mathbf{F}^{(v)}\right)^T \mathbf{F}^{(v)} = \mathbf{I}, \end{aligned} \tag{6}$$

where β is a balance parameter. Following (5), we can simply feed the average of multiple spectral embeddings, $\widehat{\mathbf{F}} = \frac{1}{m}\sum_{v=1}^{m}\mathbf{F}^{(v)}$, to the spectral clustering and obtain the discrete clustering with k-means. However, this naive way is not enough to take full advantage of rich information. Besides, the final clustering result would be unstable as k-means is sensitive to initialization Huang et al. (2021). In order to reasonably merge the heterogeneous information, we propose to obtain the clustering indicator matrix from a more geometric viewpoint. We consider the partition of each view as a perturbation of the consensus clustering $\mathbf{Y}$, and the multiple partitions can be integrated by estimating a distinct rotation $\mathbf{R}^{(v)}$ for each partition (Huang et al. 2013; Kang et al. 2018). Mathematically, we have

$$\begin{aligned} &\min_{\{\mathbf{R}^{(v)}\},\mathbf{Y}} \sum_{v=1}^{m} \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right\|_F^2 \\ &\text{s.t.} \quad \left(\mathbf{R}^{(v)}\right)^T \mathbf{R}^{(v)} = \mathbf{I}, \mathbf{Y} \in Ind, \end{aligned} \tag{7}$$

and $\boldsymbol{\mu} = [\mu^{(1)}, \ldots, \mu^{(m)}]^T$ are the weights for different views. To effectively measure the weight $\mu^{(v)}$ for each view, we adopt an inverse distance weighting strategy as follows (Huang et al. 2018; Nie et al. 2018):

$$\mu^{(v)} = \frac{1}{2\left\| \mathbf{Y} - \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right\|_F}. \tag{8}$$

To achieve from graph learning to final discrete clustering label, we combine (7) and (8) into a joint learning framework. Thus our model named as Multiple Partitions Alignment via Spectral Rotation (MPASR) can be given by

$$\begin{aligned} &\min_{\{\mathbf{S}^{(v)}\},\{\mathbf{F}^{(v)}\},\{\mathbf{R}^{(v)}\},\mathbf{Y}} \sum_{v=1}^{m} \left\{ \sum_{i,j=1}^{n} \left\| \mathbf{x}_i^{(v)} - \mathbf{x}_j^{(v)} \right\|_2^2 s_{ij}^{(v)} + \alpha \left(s_{ij}^{(v)} \right)^2 \right. \\ &\left. + \beta \mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T \mathbf{L}^{(v)} \mathbf{F}^{(v)} \right) + \gamma \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right\|_F^2 \right\} \\ &\text{s.t.} \quad \forall i, \left(\mathbf{s}_i^{(v)}\right)^T \mathbf{1} = 1, s_{ii}^{(v)} = 0, s_{ij}^{(v)} \geq 0, \left(\mathbf{R}^{(v)}\right)^T \mathbf{R}^{(v)} = \mathbf{I}, \\ &\mathbf{F}^{(v)} \in \mathbb{R}^{n \times c}, \left(\mathbf{F}^{(v)}\right)^T \mathbf{F}^{(v)} = \mathbf{I}, \mathbf{Y} \in Ind, \end{aligned} \tag{9}$$

where γ is a trade-off parameter.

We summarize the properties of the proposed model as follows.

- Unlike previous multi-view clustering methods, the proposed MPASR merges multi-view information in partition level instead of the raw feature space where the data points lie. Considering the natural space for multi-view clustering should be all partitions, it would be more favorable to carry out information fusion based on multiple partitions.
- A weight is allocated to each view by taking the clustering capacity differences of different views into consideration. According to (8), if the v-th view has a good clustering capacity, $\left\| \mathbf{Y} - \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right\|_F$ should be small, thus the corresponding weight is large. Accordingly, a small weight will be allocated to a weak view. As a result, the clustering capacity of each view is automatically and well taken care of.
- MPASR performs graph learning, spectral clustering, and partitions fusion in a mutual reinforcement manner. That is, with the input data matrix $\mathbf{X}$, MPASR will output the

final discrete cluster label $\mathbf{Y}$. Hence it is an end-to-end single-stage joint learning model. Besides, the discretization procedure is no longer required. Thus avoids the extra postprocessing step (e.g., the k-means step), which is sensitive to initialization.

- Actually, (9) not only unifies the pipeline of optimization steps but also attempts to learn optimal structured graphs for clustering. According to the spectral graph theory, the ideal graph is expected to be c-connected if there are exactly c clusters Nie et al. (2014). That is to say, the corresponding Laplacian matrix has exactly c zero eigenvalues σ_is. Note that minimizing $\sum_{i=1}^{c}\sigma_i$ is equivalent to $\min_{\mathbf{F}^T\mathbf{F}=\mathbf{I}} Tr(\mathbf{F}^T\mathbf{L}\mathbf{F})$. Hence the third term in (9) with a suitable β is able to guarantee that each graph $\mathbf{S}^v$ contains a clear clustering structure.

3.1 Optimization algorithm for (9)

In the following, we will adopt an iterative updating algorithm to optimize problem (9). We optimize it with respect to one variable while keeping others fixed. The specific steps are introduced as follows.

(1) Updating $\{\mathbf{S}^{(v)}\}$: the optimization formula for $\{\mathbf{S}^{(v)}\}$ is

$$\begin{aligned}\min_{\{\mathbf{S}^{(v)}\}}\sum_{v=1}^{m}\left\{\sum_{i,j=1}^{n}\left\|\mathbf{x}_i^{(v)}-\mathbf{x}_j^{(v)}\right\|_2^2 s_{ij}^{(v)}+\alpha\left(s_{ij}^{(v)}\right)^2+\beta\mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T\mathbf{L}^{(v)}\mathbf{F}^{(v)}\right)\right\}\\ \text{s.t.}\quad \forall i,\left(\mathbf{s}_i^{(v)}\right)^T\mathbf{1}=1,s_{ii}^{(v)}=0,s_{ij}^{(v)}\geq 0,\end{aligned}\tag{10}$$

We can see (10) is independent for each view. For a fixed specific v, we need to solve

$$\begin{aligned}\min_{\mathbf{S}^{(v)}}\sum_{i,j=1}^{n}\left\|\mathbf{x}_i^{(v)}-\mathbf{x}_j^{(v)}\right\|_2^2 s_{ij}^{(v)}+\alpha\left(s_{ij}^{(v)}\right)^2+\beta\mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T\mathbf{L}^{(v)}\mathbf{F}^{(v)}\right)\\ \text{s.t.}\quad \forall i,\left(\mathbf{s}_i^{(v)}\right)^T\mathbf{1}=1,s_{ii}^{(v)}=0,s_{ij}^{(v)}\geq 0,\end{aligned}\tag{11}$$

Note that (11) is further independent between different i, thus it can be optimized in a vector form as

$$\begin{aligned}\min_{\mathbf{s}_i^{(v)}}\sum_{j=1}^{n}\left(d_{ij}^x s_{ij}^{(v)}+\alpha\left(s_{ij}^{(v)}\right)^2+\frac{\beta}{2}d_{ij}^f s_{ij}^{(v)}\right)\\ \text{s.t.}\quad \left(\mathbf{s}_i^{(v)}\right)^T\mathbf{1}=1,s_{ii}^{(v)}=0,s_{ij}^{(v)}\geq 0,\end{aligned}\tag{12}$$

where $d_{ij}^x=\left\|\mathbf{x}_i^{(v)}-\mathbf{x}_j^{(v)}\right\|_2^2$, $d_{ij}^f=\left\|\mathbf{f}_i^{(v)}-\mathbf{f}_j^{(v)}\right\|_2^2$, and the following equation in spectral analysis is adopted

$$\mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T\mathbf{L}^{(v)}\mathbf{F}^{(v)}\right)=\frac{1}{2}\sum_{i,j=1}^{n}\left\|\mathbf{f}_i^{(v)}-\mathbf{f}_j^{(v)}\right\|_2^2 s_{ij}^{(v)}.\tag{13}$$

Denote $\mathbf{d}_i$ as a vector with its jth element being $d_{ij}=d_{ij}^x+\frac{\beta}{2}d_{ij}^f$, the optimal solution can be obtained by solving the following compact formula

$$\begin{aligned} &\min_{\mathbf{s}_i^{(v)}} \left\| \mathbf{s}_i^{(v)} + \frac{1}{2\alpha}\mathbf{d}_i \right\|_2^2 \\ &\text{s.t.} \quad \left(\mathbf{s}_i^{(v)}\right)^T \mathbf{1} = 1, s_{ii}^{(v)} = 0, s_{ij}^{(v)} \geq 0. \end{aligned} \tag{14}$$

The solution of problem (14) will be detailedly discussed in a later section, in which the parameter α can also be automatically determined.

(2) Updating $\{\mathbf{F}^{(v)}\}$: the optimization formula for $\{\mathbf{F}^{(v)}\}$ is

$$\begin{aligned} &\min_{\{\mathbf{F}^{(v)}\}} \sum_{v=1}^{m} \left\{ \beta \mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T \mathbf{L}^{(v)} \mathbf{F}^{(v)} \right) + \gamma \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2 \right\} \\ &\text{s.t.} \quad \mathbf{F}^{(v)} \in \mathbb{R}^{n \times c}, \left(\mathbf{F}^{(v)}\right)^T \mathbf{F}^{(v)} = \mathbf{I}. \end{aligned} \tag{15}$$

For a fixed specific v, we need to solve

$$\begin{aligned} &\min_{\mathbf{F}^{(v)}} \mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T \mathbf{L}^{(v)} \mathbf{F}^{(v)} \right) + \frac{\gamma \mu^{(v)}}{\beta} \left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2 \\ &\text{s.t.} \quad \mathbf{F}^{(v)} \in \mathbb{R}^{n \times c}, \left(\mathbf{F}^{(v)}\right)^T \mathbf{F}^{(v)} = \mathbf{I}. \end{aligned} \tag{16}$$

This subproblem can be solved efficiently by a iterative algorithm developed in Wen and Yin (2013).

(3) Updating $\{\mathbf{R}^{(v)}\}$: the optimization formula for $\{\mathbf{R}^{(v)}\}$ is

$$\begin{aligned} &\min_{\{\mathbf{R}^{(v)}\}} \sum_{v=1}^{m} \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2 \\ &\text{s.t.} \quad \left(\mathbf{R}^{(v)}\right)^T \mathbf{R}^{(v)} = \mathbf{I}. \end{aligned} \tag{17}$$

Similarly, for a specific v, we need to optimize

$$\begin{aligned} &\min_{\mathbf{R}^{(v)}} \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2 \\ &\text{s.t.} \quad \left(\mathbf{R}^{(v)}\right)^T \mathbf{R}^{(v)} = \mathbf{I}. \end{aligned} \tag{18}$$

(18) is essentially an orthogonal Procrustes problem, thus we can draw a closed-form solution with the following lemma.

Lemma 1 *For problem*

$$\min_{\mathbf{R}^T\mathbf{R}=\mathbf{I}} \|\mathbf{A} - \mathbf{B}\mathbf{R}\|_F^2, \tag{19}$$

the corresponding closed-form solution is $\mathbf{R}^* = \mathbf{W}\mathbf{H}^T$, *where* $\mathbf{W}$ *and* $\mathbf{H}$ *are the left and right singular vectors of* $\mathbf{B}^T\mathbf{A}$, *respectively* (*Schönemann* 1966).

According to Lemma 1, the solution of (18) is given by

$$\mathbf{R}^{(v)} = \mathbf{W}^{(v)} \left(\mathbf{H}^{(v)}\right)^T, \tag{20}$$

where $\mathbf{W}^{(v)}$ and $\mathbf{H}^{(v)}$ are the the left and right singular vectors of $\left(\mathbf{F}^{(v)}\right)^T \mathbf{Y}$.

(4) Updating $\mathbf{Y}$: the optimization formula for $\mathbf{Y}$ is

$$\min_{\mathbf{Y}\in Ind} \sum_{v=1}^{m} \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right\|_F^2 \tag{21}$$

Unfold the above equation, we have

$$\begin{aligned} &\sum_{v=1}^{m} \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right\|_F^2 \\ &= \sum_{v=1}^{m} \mu^{(v)} \left(\|\mathbf{Y}\|_F^2 + \left\| \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right\|_F^2 \right) - 2\sum_{v=1}^{m} \mu^{(v)} \mathrm{Tr}\left(\mathbf{Y}^T \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right) \\ &= \sum_{v=1}^{m} \mu^{(v)}(n+c) - 2\mathrm{Tr}\left(\mathbf{Y}^T \left(\sum_{v=1}^{m} \mu^{(v)} \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right) \right). \end{aligned} \tag{22}$$

Thus the optimal solution can be obtained by solving

$$\max_{\mathbf{Y}\in Ind} \mathrm{Tr}\left(\mathbf{Y}^T \left(\sum_{v=1}^{m} \mu^{(v)} \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right) \right). \tag{23}$$

It has a closed-form solution, i.e., $\forall i = 1, \ldots, n$,

$$\mathbf{Y}_{ij} = \begin{cases} 1, & j = \arg\max_k \left[\sum_{v=1}^{m} \mu^{(v)} \mathbf{F}^{(v)}\mathbf{R}^{(v)} \right]_k, \\ 0, & otherwise. \end{cases} \tag{24}$$

The detailed algorithm to solve the objective in (9) is summarized in Algorithm 1.

Algorithm 1: The Algorithm for MPASR

Input: Data with m views $\{\mathbf{X}^{(1)}, \mathbf{X}^{(2)}, \ldots, \mathbf{X}^{(m)}\}$, the parameters β, γ and the number of classes c;
Initialize the weight factor $w^{(v)} = \frac{1}{m}$ for each view;
Initialize the similarity graph $\mathbf{S}^{(v)}$ by solving (4) for each view;
Construct the Laplacian matrix $\mathbf{L}^{(v)}$ for each view based on $\mathbf{S}^{(v)}$;
Initialize $\mathbf{F}^{(v)}$ by solving (5) for each view, which is formed by the eigenvectors corresponding to the smallest c eigenvalues of $\mathbf{L}^{(v)}$;
Output: The discrete indicator matrix $\mathbf{Y}$.
1: **repeat**
2: Update $\mathbf{S}^{(v)}$ according to (14).
3: Update $\mathbf{F}^{(v)}$ according to (16).
4: Update $\mathbf{R}^{(v)}$ according to (20).
5: Update $\mathbf{Y}$ according to (24).
6: Update $\mu^{(v)}$ according to (8).
7: **until** converge

3.2 Solution of problem (14)

Recall that the formulation of problem (14) is

$$\begin{aligned} &\min_{\mathbf{s}_i^{(v)}} \frac{1}{2}\left\|\mathbf{s}_i^{(v)} + \frac{1}{2\alpha}\mathbf{d}_i\right\|_2^2 \\ &\text{s.t.} \quad \left(\mathbf{s}_i^{(v)}\right)^T \mathbf{1} = 1, s_{ii}^{(v)} = 0, s_{ij}^{(v)} \geq 0. \end{aligned} \tag{25}$$

The Lagrangian function of (25) can be defined as

$$\mathcal{L}\left(\mathbf{s}_i^{(v)}, \phi, \boldsymbol{\varphi}\right) = \frac{1}{2}\left\|\mathbf{s}_i^{(v)} + \frac{1}{2\alpha}\mathbf{d}_i\right\|_2^2 - \phi\left(\left(\mathbf{s}_i^{(v)}\right)^T \mathbf{1} - 1\right) - \boldsymbol{\varphi}^T \mathbf{s}_i^{(v)}, \tag{26}$$

where ϕ and $\boldsymbol{\varphi}$ are the Lagrange multipliers with respect to the corresponding constraints.

Taking the derivative with respect to $\mathbf{s}_i^{(v)}$ and setting the derivation to zero, we get

$$\mathbf{s}_i^{(v)} + \frac{1}{2\alpha}\mathbf{d}_i - \phi\mathbf{1} - \boldsymbol{\varphi} = \mathbf{0}. \tag{27}$$

The jth element of $\mathbf{s}_i^{(v)}$ in (27) can be given as

$$s_{ij}^{(v)} + \frac{1}{2\alpha}d_{ij} - \phi - \boldsymbol{\varphi}_j = 0. \tag{28}$$

Assuming that the optimal solution to (25) is $\widetilde{\mathbf{s}}_i^{(v)}$, and the corresponding Lagrange multipliers are $\widetilde{\phi}$ and $\widetilde{\boldsymbol{\varphi}}$, respectively. According to the KKT condition (Boyd and Vandenberghe 2004), we have

$$\forall j, \widetilde{s}_{ij}^{(v)} + \frac{1}{2\alpha}d_{ij} - \widetilde{\phi} - \widetilde{\boldsymbol{\varphi}}_j = 0 \tag{29}$$

$$\forall j, \widetilde{s}_{ij} \geq 0 \tag{30}$$

$$\forall j, \widetilde{\varphi}_j \geq 0 \tag{31}$$

$$\forall j, \widetilde{s}_{ij}\widetilde{\varphi}_j = 0 \tag{32}$$

Thus we can obtain the following solution

$$\widetilde{s}_{ij}^{(v)} = \left(\frac{\alpha\widetilde{\phi} - \frac{1}{2}d_{ij}}{\alpha}\right)_+. \tag{33}$$

Suppose $s_{i1}^{(v)}, s_{i2}^{(v)}, \ldots, s_{in}^{(v)}$ are ordered from small to large. In order to guarantee a sparse $\mathbf{s}_i^{(v)}$, we enforce $\mathbf{s}_i^{(v)}$ having k nonzero entries, i.e., $\widetilde{s}_{ik}^{(v)} > 0$ and $\widetilde{s}_{i,k+1}^{(v)} = 0$. Hence we obtain

$$\alpha\widetilde{\phi} - \frac{1}{2}d_{ik} > 0, \text{and } \alpha\widetilde{\phi} - \frac{1}{2}d_{i,k+1} \leq 0. \tag{34}$$

Combining (33) and the constraint $\left(\mathbf{s}_i^{(v)}\right)^T \mathbf{1} = 1$, $\widetilde{\phi}$ can be explicitly defined as

$$\widetilde{\phi} = \frac{1}{k}\left(1 + \sum_{h=1}^{k}\frac{d_{ih}}{2\alpha}\right). \tag{35}$$

Table 1 Details of computational complexity

Steps	Calculation	Complexity
Equation (14)	Update each column of $\mathbf{S}^{(v)}$	$\mathcal{O}(nc)$
Equation (16)	c eigenvectors of $\mathbf{F}^{(v)}$	$\mathcal{O}(cn^2)$
Equation (20)	$\mathbf{R}^{(i)}$	$\mathcal{O}(cn^2)$
Equation (24)	$\arg\max_k \left[\sum_{v=1}^{m} \mu^{(v)}\mathbf{F}^{(v)}\mathbf{R}^{(v)}\right]_k$	$\mathcal{O}(nc^2)$
Equation (8)	$\mu^{(v)} = \frac{1}{2\left\|\mathbf{Y}-\mathbf{F}^{(v)}\mathbf{R}^{(v)}\right\|_F}$	$\mathcal{O}(nc^2)$
Total	$\approx \mathcal{O}(n^2)$	

According to (34) and (35), we get

$$\begin{cases} \alpha > \frac{kd_{ik} - \sum_{h=1}^{k} d_{ih}}{2}, \\ \alpha \leq \frac{kd_{i,k+1} - \sum_{h=1}^{k} d_{ih}}{2}. \end{cases} \tag{36}$$

To constrain the optimal solution $\mathbf{s}_i^{(v)}$ to have exactly k nonzero entries, α can be set to

$$\alpha = \frac{kd_{i,k+1} - \sum_{h=1}^{k} d_{ih}}{2}. \tag{37}$$

Instead of directly searching the parameter α, where the value could be from zero to infinite, we pre-define the number of neighbors k. Thus the parameter searching of α can be easier since k is an integer and its value is finite.

According to (34), (35) and (37), the final optimal solution for $s_{ij}^{(v)}$ in $\mathbf{s}_i^{(v)}$ can be given as

$$\widetilde{s}_{ij}^{(v)} = \begin{cases} \frac{d_{i,k+1} - d_{ij}}{kd_{i,k+1} - \sum_{h=1}^{k} d_{ih}} & j \leq k, \\ 0 & j > k. \end{cases} \tag{38}$$

3.3 Time complexity analysis

There are five steps that mainly determine the complexity of the propose Algorithm, as shown in Algorithm 1. Recall that n, m and c respectively represent the number of data points, views and clusters. Thus the computational complexity of each step can be summarized in Table 1

Considering that $k \ll n$ and $v \ll n$ in practical, the overall Complexity is $\mathcal{O}(n^2)$.

3.4 Convergence analysis

In this subsection, we will show that our algorithm, as shown in Algorithm 1, can find a local optimal solution. As described in (14), we can find the closed-form solution of the proposed algorithm with respect to $\mathbf{S}^{(v)}$. Here we prove the convergence of Algorithm 1 under the iteration of $\mathbf{F}^{(v)}$, $\mathbf{R}^{(v)}$, and $\mathbf{Y}$. First we introduce an important lemma as follows (Nie et al. 2010; Huang et al. 2021):

Lemma 2 *For any positive real number q and t, the following inequality holds*:

$$\sqrt{q} - \frac{q}{2\sqrt{t}} \leq \sqrt{t} - \frac{t}{2\sqrt{t}}. \tag{39}$$

Proof It is obvious that inequality $(\sqrt{q} - \sqrt{t})^2 \geq 0$, thus we have

$$(\sqrt{q} - \sqrt{t})^2 \geq 0 \Rightarrow q - 2\sqrt{qt} + t \geq 0 \Rightarrow \sqrt{q} - \frac{q}{2\sqrt{t}} \leq \frac{\sqrt{t}}{2} \Rightarrow \sqrt{q} - \frac{q}{2\sqrt{t}} \leq \sqrt{t} - \frac{t}{2\sqrt{t}}$$

which completes the proof. □

Theorem 1 *In each iteration until the algorithm converges, updated variables* $\mathbf{F}^{(v)}$, $\mathbf{R}^{(v)}$, *and* $\mathbf{Y}$ *will monotonically decrease the value of the objective as follows*:

$$\begin{aligned} &\min_{\mathbf{F}^{(v)},\mathbf{R}^{(v)},\mathbf{Y}} \sum_{v=1}^{m} \left\{ \mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T \widehat{\mathbf{L}}^{(v)} \mathbf{F}^{(v)} \right) + \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2 \right\} \\ &\text{s.t.} \quad \left(\mathbf{R}^{(v)}\right)^T \mathbf{R}^{(v)} = \mathbf{I}, \mathbf{F}^{(v)} \in \mathbb{R}^{n \times c}, \left(\mathbf{F}^{(v)}\right)^T \mathbf{F}^{(v)} = \mathbf{I}, \mathbf{Y} \in Ind, \end{aligned} \tag{40}$$

where $\widehat{\mathbf{L}}^{(v)} = \frac{\beta}{\gamma} \mathbf{L}^{(v)}$.

Proof Let $\widetilde{\mathbf{F}}^{(v)}$, $\widetilde{\mathbf{R}}^{(v)}$, and $\widetilde{\mathbf{Y}}$ denote the updated $\mathbf{F}^{(v)}$, $\mathbf{R}^{(v)}$, and $\mathbf{Y}$ respectively. According to (16), (20), and (24), $\widetilde{\mathbf{F}}^{(v)}$, $\widetilde{\mathbf{R}}^{(v)}$, and $\widetilde{\mathbf{Y}}$ obviously represent the optimal solution to each corresponding subproblem. Hence we arrive at

$$\begin{aligned} &\widetilde{\mathbf{F}}^{(v)}, \widetilde{\mathbf{R}}^{(v)}, \widetilde{\mathbf{Y}} = \arg\min_{\mathbf{F}^{(v)},\mathbf{R}^{(v)},\mathbf{Y}} \sum_{v=1}^{m} \left\{ \mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T \widehat{\mathbf{L}}^{(v)} \mathbf{F}^{(v)} \right) + \mu^{(v)} \left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2 \right\} \\ &\text{s.t.} \quad \left(\mathbf{R}^{(v)}\right)^T \mathbf{R}^{(v)} = \mathbf{I}, \mathbf{F}^{(v)} \in \mathbb{R}^{n \times c}, \left(\mathbf{F}^{(v)}\right)^T \mathbf{F}^{(v)} = \mathbf{I}, \mathbf{Y} \in Ind. \end{aligned} \tag{41}$$

Combining with definition given in (8), i.e., $\mu^{(v)} = \frac{1}{2\left\|\mathbf{Y} - \mathbf{F}^{(v)}\mathbf{R}^{(v)}\right\|_F}$, we get

$$\begin{aligned} &\sum_{v=1}^{m} \left\{ \mathrm{Tr}\left(\left(\widetilde{\mathbf{F}}^{(v)}\right)^T \widehat{\mathbf{L}}^{(v)} \widetilde{\mathbf{F}}^{(v)} \right) + \frac{\left\| \widetilde{\mathbf{Y}} - \widetilde{\mathbf{F}}^{(v)} \widetilde{\mathbf{R}}^{(v)} \right\|_F^2}{2\left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F} \right\} \\ &\leq \sum_{v=1}^{m} \left\{ \mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^T \widehat{\mathbf{L}}^{(v)} \mathbf{F}^{(v)} \right) + \frac{\left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2}{2\left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F} \right\}. \end{aligned} \tag{42}$$

According to Lemma 2, we have

$$\begin{aligned} &\sum_{v=1}^{m} \left\{ \left\| \widetilde{\mathbf{Y}} - \widetilde{\mathbf{F}}^{(v)} \widetilde{\mathbf{R}}^{(v)} \right\|_F - \frac{\left\| \widetilde{\mathbf{Y}} - \widetilde{\mathbf{F}}^{(v)} \widetilde{\mathbf{R}}^{(v)} \right\|_F^2}{2\left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F} \right\} \\ &\leq \sum_{v=1}^{m} \left\{ \left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F - \frac{\left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F^2}{2\left\| \mathbf{Y} - \mathbf{F}^{(v)} \mathbf{R}^{(v)} \right\|_F} \right\}. \end{aligned} \tag{43}$$

By summing over (42) and (43) on two sides, we obtain

$$\begin{aligned}&\sum_{v=1}^{m}\left\{\mathrm{Tr}\left(\left(\widetilde{\mathbf{F}}^{(v)}\right)^{T}\widehat{\mathbf{L}}^{(v)}\widetilde{\mathbf{F}}^{(v)}\right)+\left\|\widetilde{\mathbf{Y}}-\widetilde{\mathbf{F}}^{(v)}\widetilde{\mathbf{R}}^{(v)}\right\|_{F}\right\}\\&\quad\leq\sum_{v=1}^{m}\left\{\mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^{T}\widehat{\mathbf{L}}^{(v)}\mathbf{F}^{(v)}\right)+\left\|\mathbf{Y}-\mathbf{F}^{(v)}\mathbf{R}^{(v)}\right\|_{F}\right\}.\end{aligned}\tag{44}$$

Note that the KKT conditions of (40) are equivalent to the KKT condition of the following formulation

$$\begin{aligned}&\min_{\mathbf{F}^{(v)},\mathbf{R}^{(v)},\mathbf{Y}}\sum_{v=1}^{m}\left\{\mathrm{Tr}\left(\left(\mathbf{F}^{(v)}\right)^{T}\widehat{\mathbf{L}}^{(v)}\mathbf{F}^{(v)}\right)+\left\|\mathbf{Y}-\mathbf{F}^{(v)}\mathbf{R}^{(v)}\right\|_{F}\right\},\\&\text{s.t.}\quad\left(\mathbf{R}^{(v)}\right)^{T}\mathbf{R}^{(v)}=\mathbf{I},\mathbf{F}^{(v)}\in\mathbb{R}^{n\times c},\left(\mathbf{F}^{(v)}\right)^{T}\mathbf{F}^{(v)}=\mathbf{I},\mathbf{Y}\in Ind,\end{aligned}\tag{45}$$

which completes the proof. □

4 Experiments

We evaluate the performance of the proposed MPASR by comparing it with several related state-of-the-art methods: Co-trained multi-view spectral clustering (Co-train) Kumar and Daumé (2011), Co-regularized multi-view spectral clustering (Co-reg) Kumar et al. (2012), Auto-weighted multiple graph learning (AMGL) Nie et al. (2016), Self-weighted multi-view clustering (SwMC) Nie et al. (2017), Graph learning for multi-view clustering (MVGL) Zhan et al. (2018), Weighted multi-view spectral clustering (WMSC) Zong et al. (2018), Multiview clustering via adaptively weighted procrustes (AWP) Nie et al. (2018), Multiple partitions aligned clustering (mPAC) Kang et al. (2019), Multi-view consensus graph clustering (MCGC) Zhan et al. (2019), and Graph-based multi-view clustering (GMC) Wang et al. (2020). We also conduct the classic spectral clustering algorithm (SC) Ng et al. (2002) as baseline. We run SC on each view of a data set (e.g., SC(1) means performing SC on the 1st view) to get the clustering performance of each individual view.

4.1 Data Sets

This section evaluates the performance of the proposed method on several real-world data sets: 3 source data set (3source)[1] is collected from three news sources, i.e., Reuters, BBC, and The Guardian. There are 948 news articles covering 416 different news stories. Among them, 169 news were reported in all three sources and each news was annotated with one of six topical labels. **Yale**[2] is a classical face database which contains 165 images of 15 individuals. Each individual consists of 11 images that are obtained under different configurations: center-light, left-light, happy, sad, etc. Each image is described by three types of features. **ORL** is another face database that is comprised of 400 images from 40 subjects. Each image is taken at different times under different conditions (Cao et al. 2015). There

[1] http://mlg.ucd.ie/datasets/3sources.html.

[2] http://vision.ucsd.edu/content/yale-face-database.

Table 2 Description of the data sets (dimensionality)

Dataset	Yale	3sources	ORL	Caltech20	HAR	Digit
#Instance	165	169	400	2386	7352	10000
#View	3	3	3	6	2	2
#Cluster	6	15	40	20	6	10
$\#d_1$	4096	3560	4096	48	343	784
$\#d_2$	3304	3631	3304	40	211	256
$\#d_3$	6750	3068	6750	254	–	–
$\#d_4$	–	–	–	1984	–	–
$\#d_5$	–	–	–	512	–	–
$\#d_6$	–	–	–	928	–	–

are three views of this dataset used in our experiment. Caltech101[3] is an object recognition dataset containing 101 categories of images. Each image is represented by six type of features: Gabor, Wavelet Moments, Centrist, HOG, GIST and LBP. Following Feiping Nie and Li (2017), we select a subset which contains 2386 images of 20 classes (**Caltech20**): Brain, Binocular, Camera, Car-Side, Dolla-Bill, Face, Ferry, Hedgehog, Garfield, Motorbikes, Leopards, Pagoda, Rhino, Snoopy, Stapler, Stop-Sign, Water-Lilly, Wrench, Windsor Chair and Yin-yang. Human Activity Recognition (HAR)[4] database built from the recordings of subjects performing six activities (walking, standing, lying, etc) of daily living while carrying a waist-mounted smartphone with embedded inertial sensors. It contains 7352 instances with each being recorded by the time and frequency domain variables. Handwritten digit (Digit)[5] is from two sources, i.e., MNIST Handwritten Digits and USPS Handwritten Digits. The data set consists of 10,000 samples.

All the data sets are summarized in Table 2, where d_v denotes the dimension of features in view v.

4.2 Experimental settings

We adopt seven widely-used metrics to evaluate the performance of our model: clustering accuracy (ACC), Normalized Mutual Information (NMI), Purity, Precision, Recall, F-score, and Adjusted Rand Index (ARI) (Zhan et al. 2018). Note that these measures with a higher value means a better performance.

For the compared algorithms, we use the code from authors' website with default parameters. The parameter settings of our model will be introduced in a later section. The experiments are repeated 20 times under each parameter setting with the best average results being recorded.

4.3 Clustering results

The experimental results of various methods on all data sets are reported in Tables 3, 4, 5, 6, 7, and 8. We can observe that:

[3] http://www.vision.caltech.edu/Image_Datasets/Caltech101/.
[4] https://archive.ics.uci.edu/ml/datasets/Human+Activity+Recognition+Using+Smartphones#.
[5] http://archive.ics.uci.edu/ml/datasets/Multiple+Features.

Table 3 Clustering performance (mean±standard deviation) on dataset 3sources (%)

Method	ACC	NMI	Purity	*F*-score	Precision	Recall	ARI
SC(1)	49.11 ± 2.12	41.73 ± 0.93	62.43 ± 2.47	39.24 ± 1.69	39.69 ± 2.77	38.89 ± 1.55	21.05 ± 2.75
SC(2)	51.07 ± 4.90	44.94 ± 4.40	64.91 ± 3.25	43.84 ± 3.29	48.65 ± 5.51	40.08 ± 2.89	28.84 ± 4.65
SC(3)	49.59 ± 4.52	40.56 ± 2.57	63.79 ± 1.21	40.05 ± 3.58	44.11 ± 3.99	36.72 ± 3.51	24.02 ± 4.48
Co-train	55.38 ± 3.21	**56.31 ± 4.15**	**76.80 ± 1.91**	53.96 ± 3.42	**62.47±3.59**	47.53 ± 3.54	42.40 ± 4.14
Co-reg	57.81 ± 3.69	**55.82 ± 0.31**	**75.38 ± 1.34**	54.65 ± 2.22	58.33 ± 4.11	50.73 ± 1.51	42.25 ± 3.19
AMGL	48.25 ± 6.66	46.93 ± 6.75	68.93 ± 6.96	57.29 ± 3.43	55.42 ± 3.23	51.14 ± 6.45	34.56 ± 6.71
SwMC	49.64 ± 0.00	41.81 ± 0.00	54.38 ± 0.00	35.95 ± 0.00	43.40 ± 0.00	57.48 ± 0.00	30.40 ± 0.00
MVGL	40.77 ± 0.00	36.60 ± 0.00	57.87 ± 0.00	44.17 ± 0.00	41.86 ± 0.00	48.18 ± 0.00	33.38 ± 0.00
WMSC	57.40 ± 0.28	48.80 ± 1.00	71.18 ± 0.69	50.21 ± 0.50	54.24 ± 0.40	46.74 ± 0.74	36.57 ± 0.55
AWP	54.44 ± 0.00	45.88 ± 0.00	63.31 ± 0.00	42.46 ± 0.00	38.19 ± 0.00	47.80 ± 0.00	22.42 ± 0.00
MCGC	56.80 ± 0.00	34.21 ± 0.00	65.09 ± 0.00	51.58 ± 0.00	41.21 ± 0.00	**68.93 ± 0.00**	31.72 ± 0.00
mPAC	**61.34 ± 0.90**	50.48 ± 2.77	67.06 ± 2.46	**59.28 ± 4.65**	**58.69 ± 6.65**	60.03 ± 3.33	**47.19 ± 5.49**
GMC	**69.23 ± 0.00**	54.80 ± 0.00	74.56 ± 0.00	**60.47 ± 0.00**	48.44 ± 0.00	**70.45 ± 0.00**	**44.31 ± 0.00**
MPASR	**73.37±0.00**	**66.67±0.00**	**79.88±0.00**	**67.91±0.00**	**59.23 ± 0.00**	**79.57±0.00**	**56.25±0.00**

The best, the second best, and the third best results are highlighted in bold with underline, bold, and italic, respectively

Table 4 Clustering performance (mean±standard deviation) on dataset Yale (%)

Method	ACC	NMI	Purity	*F*-score	Precision	Recall	ARI
SC(1)	59.76 ± 3.63	63.23 ± 2.93	61.15 ± 3.64	44.80 ± 3.81	43.23 ± 3.78	46.51 ± 3.92	41.08 ± 4.08
SC(2)	54.12 ± 3.69	57.13 ± 3.16	54.91 ± 3.74	37.85 ± 3.14	35.62 ± 2.87	40.40 ± 3.61	33.54 ± 3.34
SC(3)	62.06 ± 4.90	64.11 ± 3.36	62.48 ± 4.89	46.97 ± 4.34	45.29 ± 4.40	48.81 ± 4.43	43.39 ± 4.64
Co-train	55.76 ± 4.71	60.30 ± 3.23	56.73 ± 4.09	41.49 ± 4.24	39.89 ± 4.53	43.26 ± 3.97	37.52 ± 4.57
Co-reg	56.30 ± 4.15	60.49 ± 3.11	58.00 ± 3.68	42.13 ± 3.99	40.71 ± 4.27	43.68 ± 3.80	38.23 ± 4.30
AMGL	60.73 ± 3.94	62.81 ± 1.58	62.18 ± 3.17	39.97 ± 3.17	32.97 ± 4.71	51.75 ± 3.42	35.11 ± 3.77
SwMC	**65.45 ± 0.00**	**68.35 ± 0.00**	65.45 ± 0.00	47.41 ± 0.00	42.37 ± 0.00	53.82 ± 0.00	43.56 ± 0.00
MVGL	64.24 ± 0.00	64.72 ± 0.00	65.45 ± 0.00	46.42 ± 0.00	42.50 ± 0.00	51.15 ± 0.00	42.60 ± 0.00
WMSC	61.27 ± 2.47	65.06 ± 2.06	61.64 ± 2.59	48.13 ± 2.50	46.32 ± 2.62	50.11 ± 2.51	44.62 ± 2.68
AWP	63.64 ± 0.00	64.99 ± 0.00	64.24 ± 0.00	**48.76 ± 0.00**	**46.48 ± 0.00**	51.27 ± 0.00	**45.26 ± 0.00**
MCGC	**67.27 ± 0.00**	**68.92 ± 0.00**	**67.27 ± 0.00**	48.33 ± 0.00	42.44 ± 0.00	**56.12 ± 0.00**	44.47 ± 0.00
mPAC	62.42 ± 3.21	67.49 ± 2.92	**73.13±1.75**	**49.65 ± 3.41**	**52.75 ± 2.98**	45.43 ± 3.68	**46.05 ± 3.69**
GMC	**65.45 ± 0.00**	67.36 ± 0.00	66.06 ± 0.00	48.01 ± 0.00	41.88 ± 0.00	**56.24 ± 0.00**	44.10 ± 0.00
MPASR	**70.30±0.00**	**72.04±0.00**	**70.30 ± 0.00**	**56.51±0.00**	**54.30±0.00**	**58.91±0.00**	**53.57±0.00**

The best, the second best, and the third best results are highlighted in bold with underline, bold, and italic, respectively

- The performance of the multi-view clustering methods, including ours, generally outperforms that of baseline single-view method on each individual view. It is consistent with the empirical theory that the clustering performance can be effectively improved by exploring the complementary information among multiple views.

Table 5 Clustering performance (mean±standard deviation) on dataset ORL (%)

Method	ACC	NMI	Purity	*F*-score	Precision	Recall	ARI
SC(1)	66.47 ± 3.99	62.67 ± 1.50	70.85 ± 2.97	57.34 ± 3.28	53.06 ± 3.35	62.40 ± 3.42	56.27 ± 3.37
SC(2)	67.80 ± 3.53	79.86 ± 1.59	71.15 ± 2.93	62.30 ± 4.00	57.25 ± 4.91	68.24 ± 2.98	61.61 ± 4.11
SC(3)	71.50 ± 4.60	76.07 ± 1.82	75.33 ± 3.66	64.53 ± 4.48	60.35 ± 5.37	69.42 ± 3.72	63.65 ± 4.61
Co-train	63.90 ± 2.90	81.17 ± 1.66	68.02 ± 2.42	54.02 ± 3.41	50.00 ± 3.80	58.78 ± 3.17	52.87 ± 3.50
Co-reg	66.70 ± 3.19	82.19 ± 1.38	70.08 ± 2.28	56.97 ± 3.00	53.13 ± 3.26	61.44 ± 2.92	55.90 ± 3.08
AMGL	72.90 ± 2.16	85.21 ± 1.37	77.90 ± 1.78	58.58 ± 4.11	48.31 ± 5.10	74.78 ± 2.17	57.41 ± 4.27
SwMC	70.75 ± 0.00	83.31 ± 0.00	76.75 ± 0.00	43.33 ± 0.00	29.61 ± 0.00	**80.72 ± 0.00**	41.39 ± 0.00
MVGL	75.00 ± 0.00	85.44 ± 0.00	79.25 ± 0.00	48.92 ± 0.00	34.81 ± 0.00	**82.28±0.00**	47.25 ± 0.00
WMSC	**78.48 ± 1.11**	82.15 ± 0.78	**81.53 ± 0.95**	**71.61 ± 1.69**	67.64 ± 1.78	76.08 ± 1.93	60.91 ± 1.74
AWP	**79.50 ± 0.00**	**88.98±0.00**	81.25 ± 0.00	**72.76 ± 0.00**	**68.26 ± 0.00**	77.89 ± 0.00	**72.08 ± 0.00**
MCGC	77.00 ± 0.00	**87.22 ± 0.00**	**82.75 ± 0.00**	56.25 ± 0.00	42.83 ± 0.00	**81.94 ± 0.00**	54.92 ± 0.00
mPAC	67.25 ± 2.54	**85.45 ± 1.02**	80.17 ± 0.76	60.56 ± 2.71	**73.00±0.84**	51.78 ± 3.53	59.48 ± 2.81
GMC	63.25 ± 0.00	80.35 ± 0.00	71.50 ± 0.00	65.99 ± 0.00	53.21 ± 0.00	80.11 ± 0.00	**63.67 ± 0.00**
MPASR	**79.75±0.00**	84.34 ± 0.00	**85.00±0.00**	**73.42±0.00**	**67.73 ± 0.00**	**82.28±0.00**	**72.24±0.00**

The best, the second best, and the third best results are highlighted in bold with underline, bold, and italic, respectively

Table 6 Clustering performance (mean±standard deviation) on dataset Caltech20 (%)

Method	ACC	NMI	Purity	*F*-score	Precision	Recall	ARI
SC(1)	25.85 ± 0.48	27.12 ± 0.48	52.95 ± 1.03	18.59 ± 0.42	36.73 ± 1.06	12.45 ± 0.31	11.28 ± 0.46
SC(2)	28.70 ± 0.73	34.69 ± 0.48	61.83 ± 0.75	24.16 ± 1.38	49.20 ± 2.55	16.01 ± 0.97	17.58 ± 1.45
SC(3)	29.79 ± 1.43	33.76 ± 0.87	59.21 ± 0.95	23.81 ± 1.11	47.53 ± 1.90	15.88 ± 0.81	17.05 ± 1.13
SC(4)	39.58 ± 2.89	52.53 ± 1.05	**74.68 ± 1.55**	34.55 ± 2.05	**68.41 ± 3.03**	23.11 ± 1.53	28.69 ± 2.12
SC(5)	38.39 ± 1.35	48.13 ± 0.81	71.90 ± 0.91	32.69 ± 1.51	66.02 ± 1.90	21.72 ± 1.15	26.79 ± 1.52
SC(6)	34.15 ± 1.00	45.20 ± 0.82	69.33 ± 1.12	31.51 ± 1.47	63.08 ± 2.36	21.00 ± 1.06	25.45 ± 1.52
Co-train	41.46 ± 2.45	48.57 ± 0.40	71.53 ± 0.34	35.13 ± 2.65	67.73 ± 2.15	23.73 ± 2.15	29.16 ± 2.60
Co-reg	41.93 ± 3.32	53.75 ± 0.22	**74.98 ± 1.15**	35.56 ± 1.80	67.80 ± 2.26	24.13 ± 1.71	29.55 ± 1.64
AMGL	50.29 ± 1.98	52.39 ± 3.95	66.79 ± 3.40	38.55 ± 3.03	33.81 ± 4.44	45.24 ± 1.23	24.38 ± 4.69
SwMC	**54.44 ± 0.00**	44.52 ± 0.00	66.43 ± 0.00	38.03 ± 0.00	27.91 ± 0.00	**59.67 ± 0.00**	20.38 ± 0.00
MVGL	**52.10 ± 0.00**	43.78 ± 0.00	64.63 ± 0.00	37.02 ± 0.00	26.87 ± 0.00	**59.50 ± 0.00**	18.82 ± 0.00
WMSC	33.48 ± 1.09	41.51 ± 0.48	67.02 ± 0.49	30.11 ± 1.06	58.49 ± 1.84	20.27 ± 0.75	23.72 ± 1.12
AWP	51.59 ± 0.00	**57.77 ± 0.00**	73.22 ± 0.00	**53.26 ± 0.00**	**71.17±0.00**	42.56 ± 0.00	**46.78 ± 0.00**
MCGC	47.53 ± 0.00	**54.57 ± 0.00**	68.65 ± 0.00	**40.17 ± 0.00**	41.74 ± 0.00	38.71 ± 0.00	29.06 ± 0.00
mPAC	43.83 ± 2.64	53.54 ± 0.78	53.21 ± 3.48	38.89 ± 2.23	28.22 ± 2.29	**62.70±1.57**	**32.03 ± 2.12**
GMC	45.64 ± 0.00	38.46 ± 0.00	55.49 ± 0.00	34.03 ± 0.00	52.78 ± 0.00	47.28 ± 0.00	12.84 ± 0.00
MPASR	**61.53±0.00**	**62.05±0.00**	**75.27±0.00**	**61.75±0.00**	**69.07 ± 0.00**	55.83 ± 0.00	**55.23±0.00**

The best, the second best, and the third best results are highlighted in bold with underline, bold, and italic, respectively

Table 7 Clustering performance (mean±standard deviation) on dataset HAR (%)

Method	ACC	NMI	Purity	*F*-score	Precision	Recall	ARI
SC(1)	63.53 ± 6.50	**61.34 ± 0.85**	**66.33 ± 2.71**	**59.24 ± 3.93**	57.14 ± 3.73	61.69 ± 5.79	**50.58 ± 4.67**
SC(2)	43.91 ± 3.55	40.62 ± 0.84	45.92 ± 1.11	37.91 ± 0.48	36.06 ± 0.86	40.05 ± 2.16	24.47 ± 0.13
Co-train	53.78 ± 2.25	47.18 ± 1.00	54.60 ± 1.75	43.87 ± 1.09	42.53 ± 0.75	45.30 ± 1.47	32.01 ± 1.21
Co-reg	**64.51 ± 6.46**	**62.13±0.43**	**66.45 ± 2.24**	58.55 ± 2.72	**58.03 ± 3.48**	59.10 ± 1.91	50.01 ± 3.45
AMGL	37.09 ± 1.25	40.52 ± 3.50	39.30 ± 1.59	48.64 ± 1.41	33.20 ± 1.30	**91.02 ± 0.40**	31.71 ± 2.13
SwMC	37.60 ± 0.00	41.95 ± 0.00	39.85 ± 0.00	49.34 ± 0.00	33.74 ± 0.00	**91.76±0.00**	32.70 ± 0.00
MVGL	38.21 ± 0.00	42.95 ± 0.00	40.72 ± 0.00	49.23 ± 0.00	33.91 ± 0.00	89.79 ± 0.00	32.71 ± 0.00
WMSC	**66.08 ± 0.00**	**62.05 ± 0.00**	**66.88±0.00**	**59.04 ± 0.00**	**58.45 ± 0.00**	59.33 ± 0.00	**50.66 ± 0.00**
AWP	45.16 ± 0.00	43.68 ± 0.00	45.48 ± 0.00	42.01 ± 0.00	37.04 ± 0.00	48.51 ± 0.00	28.25 ± 0.00
MCGC	39.15 ± 0.00	50.07 ± 0.00	39.21 ± 0.00	48.93 ± 0.00	36.92 ± 0.00	89.88 ± 0.00	30.01 ± 0.00
mPAC	45.80 ± 0.00	40.91 ± 0.00	49.66 ± 0.00	40.83 ± 0.00	36.88 ± 0.00	45.72 ± 0.00	27.20 ± 0.00
GMC	37.54 ± 0.00	42.10 ± 0.00	39.88 ± 0.00	49.34 ± 0.00	33.76 ± 0.00	**91.68 ± 0.00**	32.71 ± 0.00
MPASR	**66.64±0.00**	59.59 ± 0.00	57.49 ± 0.00	**62.85±0.00**	**58.75±0.00**	89.42 ± 0.00	**52.41±0.00**

The best, the second best, and the third best results are highlighted in bold with underline, bold, and italic, respectively

Table 8 Clustering performance (mean±standard deviation) on dataset Digit (%)

Method	ACC	NMI	Purity	*F*-score	Precision	Recall	ARI
SC(1)	55.33 ± 4.33	47.42 ± 1.31	58.45 ± 3.46	43.37 ± 2.27	42.56 ± 2.29	44.22 ± 2.28	36.95 ± 2.54
SC(2)	49.34 ± 1.36	46.08 ± 0.96	51.88 ± 1.30	39.11 ± 0.76	38.81 ± 0.85	39.41 ± 0.69	32.29 ± 0.86
AMGL	**91.45 ± 12.32**	**94.90 ± 6.41**	93.68 ± 8.98	**91.50 ± 12.29**	87.88 ± 17.70	96.55 ± 4.27	90.38 ± 13.96
SwMC	89.87 ± 0.00	93.39 ± 0.00	89.87 ± 0.00	90.56 ± 0.00	83.05 ± 0.00	**99.56±0.00**	89.40 ± 0.00
MVGL	**99.09±0.00**	**97.25±0.00**	**99.09±0.00**	**98.19±0.00**	**98.19±0.00**	**98.19 ± 0.00**	**97.99±0.00**
WMSC	74.20 ± 0.10	66.02 ± 0.03	74.20 ± 0.10	63.12 ± 0.08	62.97 ± 0.09	63.26 ± 0.08	59.01 ± 0.09
AWP	70.28 ± 0.00	60.54 ± 0.00	70.95 ± 0.00	60.05 ± 0.00	57.00 ± 0.00	63.45 ± 0.00	55.35 ± 0.00
MCGC	27.39 ± 0.00	21.43 ± 0.00	27.45 ± 0.00	23.96 ± 0.00	13.72 ± 0.00	94.48 ± 0.00	7.89 ± 0.00
GMC	90.23 ± 0.00	89.74 ± 0.00	**98.23 ± 0.00**	89.91 ± 0.00	**90.05 ± 0.00**	94.77 ± 0.00	**91.27 ± 0.00**
MPASR	**96.74 ± 0.00**	**95.70 ± 0.00**	**96.74 ± 0.00**	**93.26 ± 0.00**	**88.24 ± 0.00**	**98.98 ± 0.00**	**91.58 ± 0.00**

The best, the second best, and the third best results are highlighted in bold with underline, bold, and italic, respectively

- Our proposed MPASR consistently has superior or comparable performance compared to the other ten multi-view clustering methods, which verifies the effectiveness of our approach. Besides, we can see that the improvement is remarkable on several data sets. Note that the results of Co-train, Co-reg, and mPAC on Digit dataset are not recorded. It may well be because the Digit dataset is too sparse, especially the first view of Digit, nearly 90% of the elements are zeros, thus these three methods cannot handle it.
- Note that AWP performs initialization and optimization process on a fixed graph, thus is unavoidably sensitive to the input affinity matrix as well as the corresponding spectral embedding matrix. While our model accomplishes the subtasks of learning

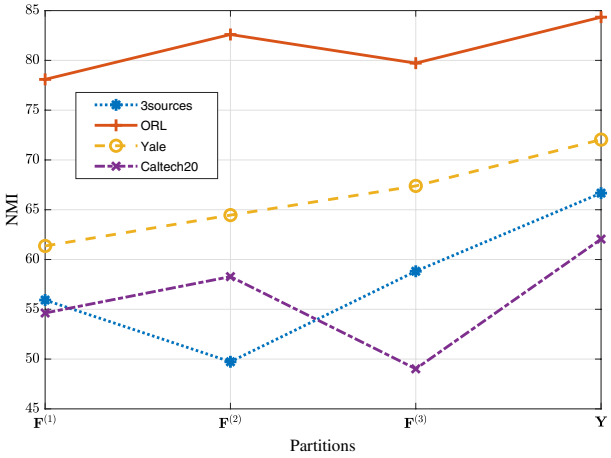

Fig. 2 Clustering performance of multiple partitions $\{\mathbf{F}^{(v)}\}|_{v=1}^{m}$ and the final consensus partition $\mathbf{Y}$

the similarity graphs adaptively, fusing the multiple partitions, and assigning cluster label to each instance in a unified framework.

- Previous studies have shown that exploring the local connectivity of data is a successful strategy for clustering task. Differ to the proposed MPASR that learns the similarity graph by exploring the local connectivity of data, mPAC focuses on learning a subspace representation by making use of the global structure of data. Note that mPAC achieves comparable clustering results several times. This inspires us that exploring both the global structure and local connectivity of data could be a good strategy to boost the clustering performance in future work.
- With respect to the compared methods that consider all graphs produce the same partition, MPASR focuses on learning one partition for each view and integrating them into a consensus one via spectral rotation. Our empirical studies demonstrate the superiority of this aligning mechanism.
- It is noteworthy that the clustering results obtained by MPASR are quite stable with the standard deviation always being zero. The reason for this is that MPASR directly outputs the discrete clustering label for each data sample, thus avoids the extra discretization procedure which is unstable and sensitive to initialization.

To better illustrate the rotation strategy of our model, we showcase the dynamics of basic partitions $\mathbf{F}^{(v)}$ as well as the consensus partition $\mathbf{Y}$ in Fig. 2. Note that the datasets 3sources, ORL, and Yale contain three views, while Caltech20 contains six views, hence we plot the first three partitions of Caltech20 for convenient illustration. We can observe that our model is able to find a good consensus partition with better clustering results, compared with other basic partitions. Despite the diverse of the basic partitions with different clustering performance, the proposed model can integrate them reasonably via the spectral rotation and achieve a a good clustering. Consequently, the proposed partition aligning mechanism is validated.

4.4 Parameter Study

For view weight parameter $\mu^{(v)}$, it can be automatically calculated according to (8). For balance parameter α, it can also be automatically determined as mentioned before. Thus

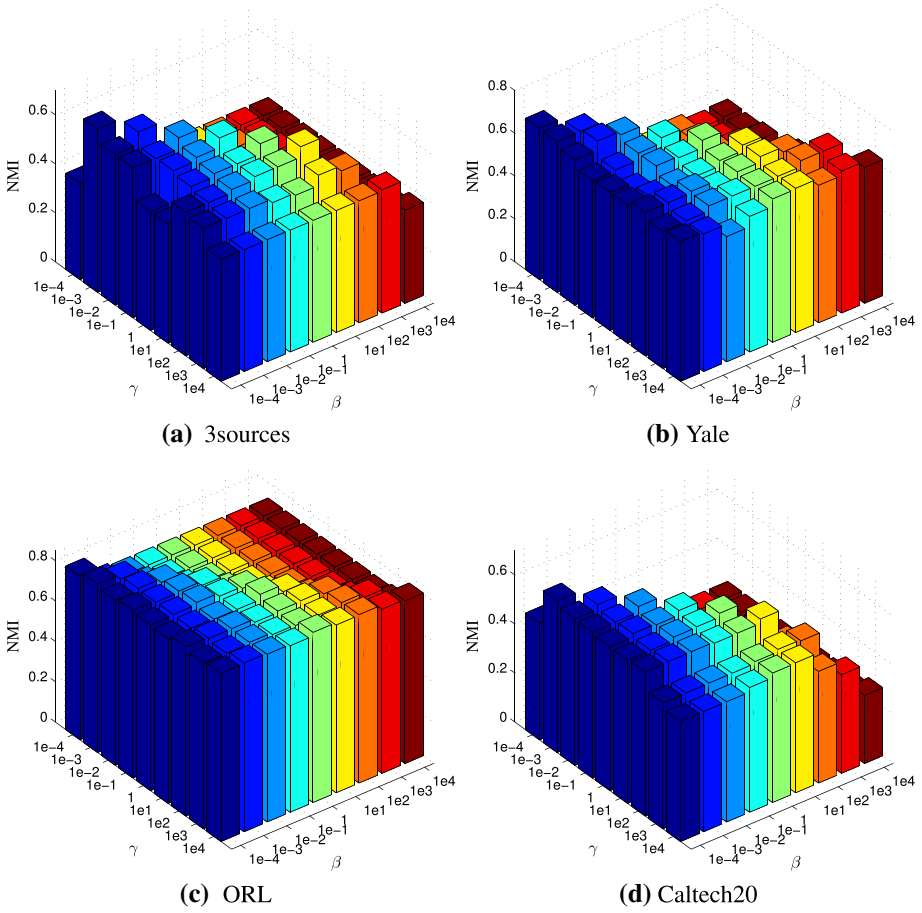

Fig. 3 The effect of parameters on datasets

we only need to tune the parameters β and γ. In this section, we set both β and γ by the grid {1e-4,1e-3,1e-2,1e-1,1,1e2,1e2,1e3,1e4} for simplicity. As shown in Figure 3, we can find that our performance is relatively stable with respect to γ under a wide range of parameter settings. As mentioned before, the ideal graph is c-connected if there are exactly c clusters (Nie et al. 2014; Huang et al. 2019). That's the possible reason why the final performance is a little sensitive to the choice of β since the third term in (9) needs to search a suitable β that is able to guarantee each graph $\mathbf{S}^{(v)}$ is optimal for clustering. Generally, MPASR obtains good performance when β varies in the range [1e-2,1e1] and γ in the range [1e-3,1e-1].

As shown in (37), parameter α can be set by the number of the nearest neighbors k, which guarantees that the optimal solution $\mathbf{s}_i^{(v)}$ to have exactly k nonzero entries. We set $k = 10$ in previous experiments for simplicity. Here we illustrate the clustering performance with respect to different k. Take the dataset 3sources as an example, we see that MPASR is a little sensitive to k, as plotted in Fig. 4. Note that obtaining an optimal k in such case is still an open problem. It is clear that the parameters of MPASR is tuned

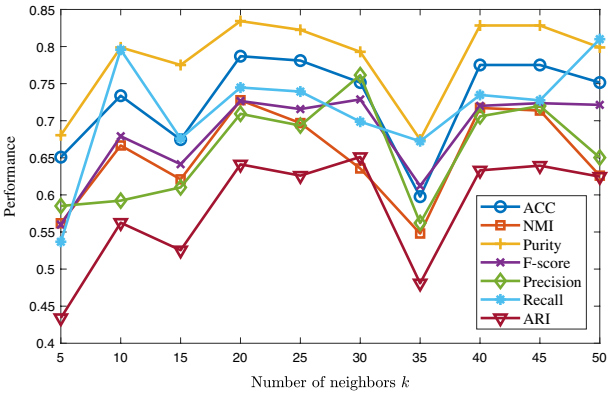

Fig. 4 Clustering performance of MPASR with respect to the number of neighbors k

Table 9 Running time of all multi-view clustering methods (seconds)

Method	Yale	3sources	ORL	Caltech20	HAR	Digit
Co-train	0.70	1.26	8.37	384.42	2539.48	–
Co-reg	1.82	1.44	6.64	198.57	529.56	–
AMGL	0.08	0.12	0.41	60.07	531.63	4054.04
SwMC	0.39	0.42	2.28	455.65	13643.60	26230.25
MVGL	0.19	0.28	0.98	248.84	4837.84	9941.77
WMSC	0.16	0.33	0.73	202.18	916.00	1153.09
AWP	0.07	0.08	0.34	63.88	612.62	1646.49
MCGC	0.12	0.14	0.59	53.14	712.13	1871.10
mPAC	1.00	6.59	10.39	11503.35	23577.02	–
GMC	0.18	0.26	0.60	34.01	606.16	1242.91
MPASR	0.65	0.96	5.51	328.53	768.63	2952.93

roughly. Better parameter tuning would achieve better clustering performance than that recorded in this paper (e.g., $k = 20$).

4.5 Computational performance

This section reports the execution time of our model as well as other compared methods. We conduct all experiments with Matlab R2018b on a machine with Core 18 Duad 2.6 GHz and 256 GB memory. As shown in Table 9, the running time of graph-based methods, including ours, is generally higher than that of other methods (e.g., AWP) due to the extra construction of the graph similarity. We see that mPAC, SwMC and MVGL are the top three time-consuming methods, whereas the execution time of AWP, GMC, and WMSC is the lowest three among the eleven multi-view clustering methods. On the whole, the proposed MPASR is faster than mPAC, SwMC and MVGL, slower then AWP and GMC but comparable with other methods, which showcases the effectiveness of MPASR. As mentioned before, the results of Co-train, Co-reg, and mPAC on Digit dataset are not recorded probably because of the high sparsity of this dataset. Hence we do not report the running time of Co-train, Co-reg, and mPAC on Digit.

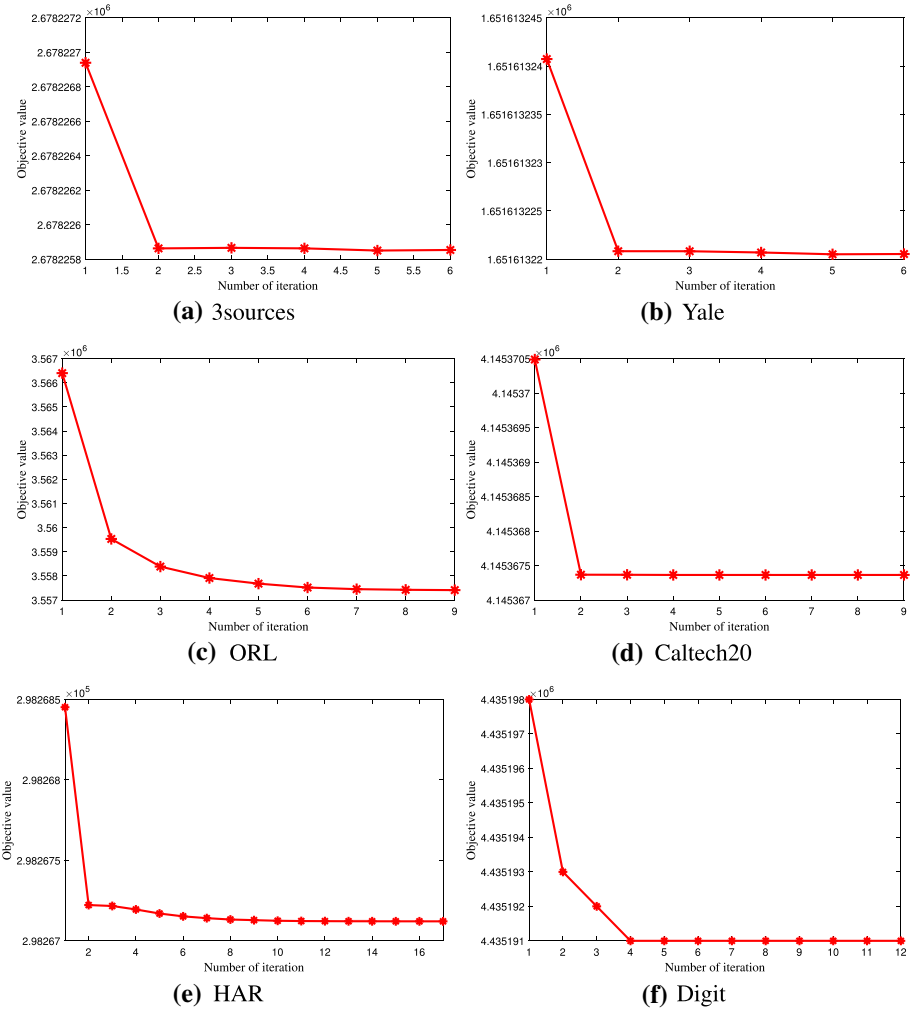

Fig. 5 The convergence of our model on datasets

4.6 Convergence analysis

The proposed iterative updating algorithm for MPASR is iterative. Here we study how fast the proposed algorithm can converge. The stop criteria for our model is defined as $|O^{t+1} - O^t|/O^t \leq 10^{-6}$ where O^t denotes the objective value in the tth iteration. We plot the convergence curves of MPASR in Fig. 5, where the x-axis means the number of iteration and y-axis denotes the objective value. It is clear that the updating rules for our model converge very fast, usually within 20 iterations, which empirically verifies the efficiency of the proposed optimization algorithm. Note that the inputs of the proposed Algorithm 1, $\mathbf{S}^{(v)}$ and $\mathbf{F}^{(v)}$, are well initialized for each view by solving (4) and (5) respectively. That's the possible reason why the proposed optimization algorithm only needs a few iterations to converge. Particularly for some small-scale datasets, e.g., 3sources and Yale, about two or three iterations are enough to reach the convergence.

5 Conclusion

In this paper, we propose to merge multi-view information in partition level instead of the raw feature space where the data points lie. The partition of each view is treated as a perturbation of the consensus clustering, and the multiple partitions are integrated by estimating a distinct rotation for each partition. We model the partitions fusion following two intuitive principles: (1) each partition can be essentially seen as a perturbation of the consensus clustering; and (2) the closer a partition to the consensus clustering, the larger weight should be allocated. Furthermore, the weight of each partition can be determined automatically. The proposed model is formulated as a joint learning framework, i.e., with the input data matrix, our model directly outputs the final discrete clustering result. Hence it is an end-to-end single-stage learning model. An iterative updating algorithm is proposed to solve the learning problem, in which the involved variables can be optimized in a mutual reinforcement manner. Finally, the experimental results have shown that (1) MPASR has superior performance compared to the other multi-view clustering methods which demonstrates the superiority of this aligning mechanism; (2) the proposed model is relatively stable across a wide range of parameter settings which showcases the robustness of the proposed model; and (3) the designed optimization algorithm is very efficient and converges fast. In the future, we are interested in extending the proposed aligning mechanism to other machine learning framework such as semi-supervised learning and deep learning.

Acknowledgements We thank the anonymous associate editor and reviewers for their helpful comments and suggestions.

Author Contributions Shudong Huang and Ivor W. Tsang carried out the theoretical analysis and experimental studies. Shudong Huang and Zenglin Xu were the major contributors in writing the manuscript. Jiancheng Lv led this project and helped to draft the manuscript. All authors read and approved the final manuscript.

Funding This work was partially supported by the National Natural Science Fund for Distinguished Young Scholar under Grant 61625204, the State Key Program of the National Science Foundation of China under Grant 61836006, Sichuan Science and Technology Program under Grant 2020YFG0323, the Fundamental Research Funds for the Central Universities under Grant 1082204112364, and ARC grant DP180100106 and DP200101328.

Data Availability Statement All data analysed during this study are included in this manuscript.

Declarations

Conflict of interest The authors declare that they have no competing interests.

References

Boyd, S., & Vandenberghe, L. (2004). *Convex optimization*. Cambridge University Press.

Cao, X., Zhang, C., Fu, H., Liu, S., & Zhang, H. (2015). Diversity-induced multi-view subspace clustering. In *Proceedings of the 28th IEEE conference on computer vision and pattern recognition* (pp. 586–594).

Du, L., & Shen, Y.D. (2015). Unsupervised feature selection with adaptive structure learning. In *Proceedings of the 21th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 209–218).

Feiping Nie, G.C., & Li, X. (2017). Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of 31st AAAI conference on artificial intelligence* (pp. 2408–2414).

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

He, R., Zheng, W. S., Tan, T., & Sun, Z. (2013). Half-quadratic-based iterative minimization for robust sparse representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 36*(2), 261–275.

Hong, C., Yu, J., You, J., Chen, X., & Tao, D. (2015). Multi-view ensemble manifold regularization for 3d object recognition. *Information Sciences, 320,* 395–405.

Huang, J., Nie, F., & Huang, H. (2013). Spectral rotation versus k-means in spectral clustering. In *Proceedings of the 27th AAAI conference on artificial intelligence* (pp. 431–437). Citeseer.

Huang, J., Nie, F., & Huang, H. (2015). A new simplex sparse learning model to measure data similarity for clustering. In *IJCAI* (pp. 3569–3575).

Huang, S., Kang, Z., Tsang, I. W., & Xu, Z. (2019). Auto-weighted multi-view clustering via kernelized graph learning. *Pattern Recognition, 88*, 174–184.

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., & Liu, Q. (2021). Robust deep k-means: an effective and simple method for data clustering. *Pattern Recognition, 117,* 1–10.

Huang, S., Tsang, I., Xu, Z., & Lv, J.C. (2021). Measuring diversity in graph learning: a unified framework for structured multi-view clustering. IEEE Transactions on Knowledge and Data Engineering. pp. 1–15.

Huang, S., Xu, Z., Tsang, I. W., & Kang, Z. (2020). Auto-weighted multi-view co-clustering with bipartite graphs. *Information Sciences, 512*, 18–30.

Kang, Z., Guo, Z., Huang, S., Wang, S., Chen, W., Su, Y., & Xu, Z. (2019). Multiple partitions aligned clustering In *Proceedings of the twenty-eighth international joint conference on artificial intelligence* (pp. 2701–2707).

Kang, Z., Peng, C., Cheng, Q., & Xu, Z. (2018). Unified spectral clustering with optimal graph. In *Proceedings of the 30th AAAI conference on artificial intelligence* (pp. 3366–3373).

Kumar, A., & Daumé, H. (2011). A co-training approach for multi-view spectral clustering. In *Proceedings of the 28th international conference on machine learning* (pp. 393–400).

Kumar, A., Rai, P., & Daume, H. (2012). Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems* (pp. 1413–1421).

Li, J. H., Wang, C. D., Li, P. Z., & Lai, J. H. (2018). Discriminative metric learning for multi-view graph partitioning. *Pattern Recognition, 75*, 199–213.

Lin, Y., Gou, Y., Liu, Z., Li, B., Lv, J., & Peng, X. (2021). Completer: incomplete multi-view clustering via contrastive prediction. In *IEEE conference on computer vision and pattern recognition* (pp. 11,174–11,183).

Lu, C., Feng, J., Lin, Z., Mei, T., & Yan, S. (2018). Subspace clustering by block diagonal representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 41*(2), 487–501.

Ma, G., He, L., Lu, C.T., Shao, W., Yu, P.S., Leow, A.D., & Ragin, A.B. (2017). Multi-view clustering with graph embedding for connectome analysis. In *Proceedings of the 2017 ACM on conference on information and knowledge management* (pp. 127–136).

Ng, A.Y., Jordan, M.I., & Weiss, Y. (2002). On spectral clustering: analysis and an algorithm. In *Advances in neural information processing systems* (pp. 849–856).

Nie, F., Huang, H., Cai, X., & Ding, C.H. (2010). Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization. In *Advances in neural information processing systems* (pp. 1813–1821).

Nie, F., Li, J., & Li, X. (2016). Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In: *Proceedings of the twenty-fifth international joint conference on artificial intelligence* (pp. 1881–1887).

Nie, F., Li, J., & Li, X. (2017). Self-weighted multiview clustering with multiple graphs. In *Proceedings of the twenty-sixth international joint conference on artificial intelligence* (pp. 2564–2570).

Nie, F., Tian, L., & Li, X. (2018). Multiview clustering via adaptively weighted procrustes. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 2022–2030).

Nie, F., Wang, X., & Huang, H. (2014). Clustering and projected clustering with adaptive neighbors. In: *Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining* (pp. 977–986). ACM.

Schönemann, P. H. (1966). A generalized solution of the orthogonal procrustes problem. *Psychometrika, 31*(1), 1–10.

Sun, S. (2013). A survey of multi-view machine learning. *Neural Computing and Applications, 23*(7–8), 2031–2038.

Wang, H., Yang, Y., & Liu, B. (2020). Gmc: graph-based multi-view clustering. *IEEE Transactions on Knowledge and Data Engineering, 32*(6), 1116–1129.

Wang, R., Nie, F., Wang, Z., Hu, H., & Li, X. (2019). Parameter-free weighted multi-view projected clustering with structured graph learning. IEEE Transactions on Knowledge and Data Engineering, pp. 1–12.

Wen, Z., & Yin, W. (2013). A feasible method for optimization with orthogonality constraints. *Mathematical Programming, 142*(1–2), 397–434.

Yang, L., Shen, C., Hu, Q., Jing, L., & Li, Y. (2019). Adaptive sample-level graph combination for partial multiview clustering. *IEEE Transactions on Image Processing, 29*, 2780–2794.

Yang, M., Li, Y., Huang, Z., Liu, Z., Hu, P., & Peng, X. (2021). Partially view-aligned representation learning with noise-robust contrastive loss. In *IEEE conference on computer vision and pattern recognition* (pp. 1134–1143).

Zhan, K., Nie, F., Wang, J., & Yang, Y. (2019). Multiview consensus graph clustering. *IEEE Transactions on Image Processing, 28*(3), 1261–1270.

Zhan, K., Zhang, C., Guan, J., & Wang, J. (2018). Graph learning for multiview clustering. IEEE Transactions on Cybernetics pp. 2887–28,959.

Zhang, C., Liu, Y., & Fu, H. (2019). Ae2-nets: autoencoder in autoencoder networks. In *IEEE conference on computer vision and pattern recognition* (pp. 2577–2585).

Zhang, Y., Jiang, Z., & Davis, L.S. (2013). Learning structured low-rank representations for image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 676–683).

Zong, L., Zhang, X., Liu, X., & Yu, H. (2018). Weighted multi-view spectral clustering based on spectral perturbation. In: *Proceedings of 32nd AAAI conference on artificial intelligence* (pp. 4621–4628).