Multiple Partitions Aligned Clustering

Zhao Kang[1] , **Zipeng Guo**[1] , **Shudong Huang**[1] , **Siying Wang**[1] ,
Wenyu Chen[1] , **Yuanzhang Su**[2*] and **Zenglin Xu**[1]
[1]School of Computer Science and Engineering,
University of Electronic Science and Technology of China
[2]School of Foreign Languages, University of Electronic Science and Technology of China
zkang@uestc.edu.cn, {zpguo, huangsd, siyingwang}@std.uestc.edu.cn, {cwy, syz, zlx}@uestc.edu.cn

Abstract

Multi-view clustering is an important yet challenging task due to the difficulty of integrating the information from multiple representations. Most existing multi-view clustering methods explore the heterogeneous information in the space where the data points lie. Such common practice may cause significant information loss because of unavoidable noise or inconsistency among views. Since different views admit the same cluster structure, the natural space should be all partitions. Orthogonal to existing techniques, in this paper, we propose to leverage the multi-view information by fusing partitions. Specifically, we align each partition to form a consensus cluster indicator matrix through a distinct rotation matrix. Moreover, a weight is assigned for each view to account for the clustering capacity differences of views. Finally, the basic partitions, weights, and consensus clustering are jointly learned in a unified framework. We demonstrate the effectiveness of our approach on several real datasets, where significant improvement is found over other state-of-the-art multi-view clustering methods.

1 Introduction

As an important problem in machine learning and data mining, clustering has been extensively studied for many years [Jain, 2010]. Technology advances have produced large volumes of data with multiple views. Multi-view features depict the same object from different perspectives, thereby providing complementary information. To leverage the multi-view information, multi-view clustering methods have drawn increasing interest in recent years [Chao *et al.*, 2017]. Due to its unsupervised learning nature, multi-view clustering is still a challenging task. The key question is how to reach a consensus of clustering among all views.

In the clustering field, two dominating methods are k-means [Jain, 2010] and spectral clustering [Ng *et al.*, 2002]. Numerous variants of them have been developed over the past decades [Chen *et al.*, 2013; Liu *et al.*, 2018; Yang *et al.*, 2018; Kang *et al.*, 2018a]. Among them, some can tackle multi-view data, e.g., multi-view kernel k-means (MKKM) [Tzortzis and Likas, 2012], robust multi-view kernel k-means (RMKKM) [Cai *et al.*, 2013], Co-trained multi-view spectral clustering (Co-train) [Kumar and Daumé, 2011], Co-regularized multi-view spectral clustering (Co-reg) [Kumar *et al.*, 2011]. Along with the development of nonnegative matrix factorization (NMF) technique, multi-view NMF also gained a lot of attention. For example, a multi-manifold regularized NMF (MNMF) is designed to preserve the local geometrical structure of the manifolds for multi-view clustering [Zong *et al.*, 2017].

Recently, subspace clustering method has shown impressive performance. Subspace clustering method first obtains a graph, which reveals the relationship between data points, then applies spectral clustering to achieve the embedding of original data, finally utilizes k-means to obtain the final clustering result [Elhamifar and Vidal, 2013; Kang *et al.*, 2019a]. Inspired by it, subspace clustering based multi-view clustering methods [Gao *et al.*, 2015; Zhang *et al.*, 2017; Huang *et al.*, 2019] have become popular in recent years. For instance, Gao et al. proposed multi-view subspace clustering (MVSC) method [Gao *et al.*, 2015]. In this approach, multiple graphs are constructed and they are forced to share the same cluster pattern. Therefore, the final clustering is a negotiated result and it might not be optimal. [Wang *et al.*, 2016] supposes that each graph should be close to each other. After obtaining graphs, their average is utilized to perform spectral clustering. The averaging strategy might be too simple to fully take advantage of heterogeneous information. Furthermore, it is a two-stage algorithm. The constructed graph might not be optimal for the subsequent clustering [Kang *et al.*, 2017].

By contrast, another class of graph-based multi-view clustering method learns a common graph based on adaptive neighbors idea [Nie *et al.*, 2016a; Zhan *et al.*, 2017]. In specific, x_i is connected to x_j with probability s_{ij}. s_{ij} should have a large value if the distance between x_i and x_j is small. Otherwise, s_{ij} should be small. Therefore, obtained s_{ij} is treated as the similarity between x_i and x_j. In [Nie *et al.*, 2016a], each view shares the same similarity graph. Moreover, a weight for each view is automatically assigned based on loss value. Though this approach has shown its competitiveness, one shortcoming of it is that it fails to consider the

*Corresponding author

flexible local manifold structures of different views.

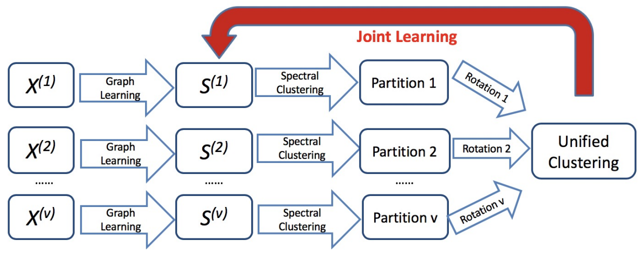


Figure 1: Illustration of our mPAC. mPAC integrates graph learning, spectral clustering, and consensus clustering into a unified framework.

Although proved to be effective in many cases, existing graph-based multi-view clustering methods are limited in several aspects. First, they integrate the multi-view information in the feature space via some simple strategies. Due to the generally unavoidable noise in the data representation, the graphs might be severely damaged and cannot represent the true similarities among data points [Kang *et al.*, 2019b]. It would make more sense if we directly reach consensus clustering in partition space where a common cluster structure is shared by all views, while the graphs might be quite different for different views. Hence, partitions from various views might be less affected by noise and easier to reach an agreement. Second, most existing algorithms follow a multi-stage strategy, which might degrade the final performance. For example, the learned graph might not be suitable for the subsequent clustering task. A joint learning method is desired for this kind of problem.

Regarding the problems mentioned above, we propose a novel multiple Partitions Aligned Clustering (mPAC) method. Fig. 1 shows the idea of our approach. mPCA performs graph construction, spectral embedding, and partitions integration via joint learning. In particular, an iterative optimization strategy allows the consensus clustering to guide the graph construction, which later contributes to a new unified clustering. To sum up, we have our two-fold contributions as follows:

- Orthogonal to existing multi-view clustering methods, we integrate multi-view information in partition space. This change in paradigm accompanies several benefits.
- An end-to-end single stage model is developed to achieve from graph construction to final clustering. Especially, we assume that the unified clustering is reachable for each view through a distinct transformation. Moreover, the output of our algorithm is the discrete cluster indicator matrix, thus no more subsequent step is needed.

Notations In this paper, matrices and vectors are represented by capital and lower-case letters, respectively. For $A = [a_{ij}] \in \mathbb{R}^{m\times n}$, $A_{i,:}$ and $A_{:,j}$ represents the i-th row and j-th column of A, respectively. The ℓ_2-norm of vector x is defined as $\|x\| = \sqrt{x^T \cdot x}$, where T means transpose. $Tr(A)$ denotes the trace of A. $\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$ denotes the Frobenius norm of A. Vector $\mathbf{1}$ indicates its elements are all ones. I refers to the identity matrix with a proper size. $Ind \overset{\text{def}}{=} \{Y \in \{0,1\}^{n\times c} | Y\mathbf{1} = \mathbf{1}\}$ represents the set of indicator matrices. We use the superscript A^i or subscript A_i to denote the i-th view of A interchangeably when convenient.

2 Subspace Clustering Revisited

In general, for data $X \in \mathcal{R}^{m\times n}$ with m features and n samples, the popular subspace clustering method can be formulated as:

$$\min_S \|X - XS\|_F^2 + \alpha\mathcal{R}(S) \quad s.t. \quad diag(S) = 0, \tag{1}$$

where $\alpha > 0$ is a balance parameter and $\mathcal{R}(Z)$ is some regularization function, which varies in different algorithms [Peng *et al.*, 2018]. For simplicity, we just apply the Frobenius norm in this paper. $diag(S)$ is the vector consists of diagonal elements of S. S is treated as the affinity graph. Therefore, once S is obtained, we can implement spectral clustering algorithm to obtain the clustering results, i.e.,

$$\min_F Tr(F^T LF) \ \ s.t. \ \ F^T F = I, \tag{2}$$

where $L \in \mathcal{R}^{n\times n}$ is the Laplacian of graph S and $F \in \mathbb{R}^{n\times c}$ is the spectral embedding and c is number of clusters. Graph Laplacian L is defined by $L = D - S$, where D is a diagonal matrix with $d_{ii} = \sum_j s_{ij}$. Since F is not discrete, k-means is often used to recover the indicator matrix $Y \in Ind$.

When data of multiple views are available, Eq. (2) can be extended to this scenario accordingly. $X = [X^1; X^2; \cdots; X^v] \in \mathbb{R}^{m\times n}$ denotes the data with v views, where $X^i \in \mathbb{R}^{m_i\times n}$ represents the i-th view data with m_i features. Basically, most methods in the literature solve the following problem

$$\min_{S,S^i} \sum_i \|X^i - X^i S^i\|_F^2 + \alpha G(S, S^i) \ s.t. \ diag(S^i) = 0, \tag{3}$$

where G represents some strategy to obtain a consensus graph S. For example, [Gao *et al.*, 2015] enforces each graph to share the same F; [Wang *et al.*, 2016] penalizes the discrepancy between graphs, then their average is used as input to spectral clustering.

We observe that there are several drawbacks shared by these approaches. First and foremost, they still lack an effective way to integrate multi-view knowledge while simultaneously considering the heterogeneity among views. Simply taking the average of graphs or assigning a unique spectral embedding is not enough to take full advantage of rich information. The graph representation itself might not be optimal to characterize the multi-view information. Secondly, they adopt a multi-stage approach. Since there is no mechanism to ensure the quality of learned graphs, this approach might lead to sub-optimal clustering results, which often occurs when noise exists. To address the above-mentioned challenging issues, we propose a multiple Partitions Aligned Clustering (mPAC) method.

3 Proposed Approach

Unlike Eq.(3), which learns a unique graph based on multiple graphs S^is, we propose to learn a partition for each graph. In specific, we adopt a joint learning strategy and formulate our objective function as

$$\min_{S^i, F_i} \sum_{i=1}^{v} \left\{ \|X^i - X^i S^i\|_F^2 + \alpha \|S^i\|_F^2 + \beta Tr(F_i^T L^i F_i) \right\}$$
$$s.t.\ \ diag(S^i) = 0,\ F_i^T F_i = I. \quad (4)$$

Next, we propose a way to fuse the multi-view information in the partition space. For multi-view clustering, a shared cluster structure is assumed. It is reasonable to assume a cluster indicator matrix $Y \in Ind$ for all views. Unfortunately, F_i's elements are continuous. The discrepancy also exists among F_i's. Thus, it is challenging to integrate multiple F_is. To recover the underlying cluster Y, we assume that each partition is a perturbation of Y and it can be aligned with Y through a rotation [Kang *et al.*, 2018b; Nie *et al.*, 2018]. Mathematically, it can be formulated as

$$\min_{Y, R_i} \sum_{i=1} \|Y - F_i R_i\|_F^2 \ \ s.t.\ \ Y \in Ind,\ R_i^T R_i = I, \quad (5)$$

where R_i represents an orthogonal matrix. Eq. (5) treats each view equally. As shown by many researchers, it is necessary to distinguish their contributions. Therefore, we introduce a weight parameter w_i for view i. Deploying a unified framework, we eventually reach our objective for mPAC as

$$\min_{S^i, F_i, Y, w_i, R_i} \sum_{i=1}^{v} \left\{ \|X^i - X^i S^i\|_F^2 + \alpha \|S^i\|_F^2 + \right.$$
$$\left. \beta Tr(F_i^T L^i F_i) + \frac{\gamma}{w_i} \|Y - F_i R_i\|_F^2 \right\} \quad (6)$$
$$s.t.\ \ diag(S^i) = 0,\ F_i^T F_i = I,\ Y \in Ind,$$
$$R_i^T R_i = I,\ w_i \geq 0,\ w\mathbf{1} = 1.$$

We can observe that the proposed approach is distinct from other methods in several aspects:

- Orthogonal to existing multi-view clustering techniques, Eq. (6) integrates heterogeneous information in partition space. Considering that a common cluster structure is shared by all views, it would be natural to perform information fusion based on partitions.
- Generally, learning with multi-stage strategy often leads to sub-optimal performance. We adopt a joint learning framework. The learning of similarity graphs, spectral embeddings, view weights, and unified cluster indicator matrix is seamlessly integrated together.
- Y is the final discrete cluster indicator matrix. Hence, discretization procedure is no longer needed. This eliminates the k-means post-processing step, which is sensitive to initialization. With input X, (6) will output the final discrete Y. Thus, it is an end-to-end single-stage learning problem.
- Multiple graphs are learned in our approach. Hence, the local manifold structures of each view are well taken care of.
- As a matter of fact, Eq. (6) is not a simple unification of the pipeline of steps and it attempts to learn graphs with optimal structure for clustering. According to the graph spectral theory, the ideal graph is c-connected if there are c clusters [Kang *et al.*, 2018b]. In other words, the Laplacian matrix L has c zero eigenvalues σ_is. Approximately, we can minimize $\sum_{i=1}^{c} \sigma_i$, which is equivalent to $\min_{F^T F = I} Tr(F^T L F)$. Hence, the third term in Eq. (6) ensures that each graph S^i is optimal for clustering.

4 Optimization Methods

To handle the objective function in Eq. (6), we apply an alternating minimization scheme to solve it.

4.1 Update S^i for Each View

By fixing other variables, we solve S^i according to

$$\min_{S^i} \sum_{i=1}^{v} \left\{ \|X^i - X^i S^i\|_F^2 + \alpha \|S^i\|_F^2 + \beta Tr(F_i^T L^i F_i) \right\}$$
$$s.t.\ \ diag(S^i) = 0. \quad (7)$$

It can be seen that each S^i is independent from other views. Therefore, we can solve each view separately. To simplify the notations, we ignore the view index tentatively. Note that L is a function of S and $Tr(F^T L F) = \sum_{ij} \frac{1}{2} \|F_{i,:} - F_{j,:}\|^2 S_{ij}$. Equivalently, we solve

$$\min_{S_{:,i}} \|X_{:,i} - X S_{:,i}\|^2 + \alpha S_{:,i}^T S_{:,i} + \frac{\beta}{2} h_i^T S_{:,i}, \quad (8)$$

where $h_i \in \mathcal{R}^{n \times 1}$ with the j-th component defined by $h_{ij} = \|F_{i,:} - F_{j,:}\|^2$. By setting its first-order derivative to zero, we obtain

$$S_{:,i} = (\alpha I + X^T X)^{-1} \Big[(X^T X)_{i,:} - \frac{\beta h_i}{4} \Big]. \quad (9)$$

4.2 Update F_i for Each View

Similarly, we drop all unrelated terms with respect to F_i and ignore the view indexes. It yields,

$$\min_{F} \beta Tr(F^T L F) + \frac{\gamma}{w_i} \|Y - FR\|_F^2 \ \ s.t.\ F^T F = I. \quad (10)$$

This sub-problem can be efficiently solved based on the method developed in [Wen and Yin, 2013].

4.3 Update R_i for Each View

With respect to R_i, the objective function is additive. We can solve each R_i individually. Specifically,

$$\min_{R} \|Y - FR\|_F^2 \ \ s.t.\ R^T R = I. \quad (11)$$

Lemma 1. *For problem*

$$\min_{R^T R = I} \|Y - FR\|_F^2, \quad (12)$$

its closed-form solution is $R^* = UV^T$, where U, V are the left and right unitary matrix of the SVD decomposition of $F^T Y$, respectively [Schönemann, 1966].

Data	Handwritten	Caltech7	Caltech20	BBCSport
View #	6	6	6	4
Points	2000	1474	2386	116
Cluster #	10	7	20	5

Table 1: Description of the data sets.

4.4 Update Y

For Y, we get

$$\min_{Y} \sum_{i=1}^{v} \frac{1}{w_i} \|Y - F_i R_i\|_F^2 \quad s.t.\ Y \in Ind. \tag{13}$$

Let's unfold above objective function, we have

$$\begin{aligned} & \sum_{i=1}^{v} \frac{1}{w_i} \|Y - F_i R_i\|_F^2 \\ = & \sum_{i=1}^{v} \frac{1}{w_i} (\|Y\|_F^2 + \|F_i R_i\|_F^2) - \sum_{i=1}^{v} \frac{2}{w_i} Tr(Y^T F_i R_i) \\ = & \sum_{i=1}^{v} \frac{n+c}{w_i} - 2Tr\Big(Y^T (\sum_{i=1}^{v} \frac{F_i R_i}{w_i})\Big). \end{aligned}$$

Thus, we can equivalently solve

$$\max_{Y \in Ind} Tr\Big(Y^T (\sum_{i=1}^{v} \frac{F_i R_i}{w_i})\Big). \tag{14}$$

It admits a closed-form solution, that is, $\forall i = 1, \cdots, n$,

$$Y_{ij} = \begin{cases} 1 & j = \arg\max_k [\sum_{i=1}^{v} \frac{F_i R_i}{w_i}]_k, \\ 0 & \text{otherwise.} \end{cases} \tag{15}$$

4.5 Update w_i for Each View

Let's denote $\|Y - F_i R_i\|_F$ as q_i, then this subproblem can be expressed as

$$\min_{w_i \geq 0, w\mathbf{1}=1} \sum_{i=1}^{v} \frac{q_i^2}{w_i}. \tag{16}$$

Based on Cauchy-Schwarz inequality, we have

$$\sum_{i=1}^{v} \frac{q_i^2}{w_i} = \Big(\sum_{i=1}^{v} \frac{q_i^2}{w_i}\Big)\Big(\sum_{i=1}^{v} w_i\Big) \geq \Big(\sum_{i=1}^{v} q_i\Big)^2. \tag{17}$$

The minimum, which is a constant, is achieved when $\sqrt{w_i} \propto \frac{q_i}{\sqrt{w_i}}$. Thus, the optimal w is given by, $\forall i = 1, \cdots, v$,

$$w_i = \frac{q_i}{\sum_{i=1}^{v} q_i}. \tag{18}$$

.

For clarity, we summarize the algorithm[1] to solve Eq. (6) in Algorithm 1.

5 Experiments

5.1 Experimental Setup

We conduct experiments on four benchmark data sets: BBC-Sport, Caltech7, Caltech20, Handwritten Numerals. Their statistics information is summarized in Table 1. We compare the proposed mPAC with several state-of-the-art methods from different categories, including Co-train [Kumar and Daumé, 2011], Co-reg [Kumar *et al.*, 2011], MKKM [Tzortzis and Likas, 2012], RMKM [Cai *et al.*, 2013], MVSC [Gao *et al.*, 2015], MNMF [Zong *et al.*, 2017], parameter-free auto-weighted multiple graph learning (AMGL) [Nie *et al.*, 2016a]. Furthermore, the classical k-means (KM) method with concatenated features (i.e., all features, AllFea in short) is included as a baseline. That is to say, all views are of the same importance. Following [Huang *et al.*, 2018], all values of each view are normalized into range $[-1, 1]$. To achieve a comprehensive evaluation, we apply five widely-used metrics to examine the effectiveness of our method: F-score, precision, Recall, Normalized Mutual Information (NMI), and Adjusted Rand Index (ARI). We initialize our algorithm by using the results from [Nie *et al.*, 2016b].

Algorithm 1 Optimization for mPAC

Input: Multiview matrix $X^1, \cdots, X^v$, cluster number c, parameters α, β, γ.
Output: Y.
Initialize: Random Y and F_i, $R_i = I$, $w_i = 1/v, \forall i = 1, \cdots, v$.
REPEAT

```
for view 1 to v do
    Update each column of S according to (9);
    Solve the subproblem (10);
    Solve the subproblem (11);
end for
Update Y according to (15);
Update w_i via (18) for each view.
```

UNTIL stopping criterion is met

5.2 Experimental Results

We repeat each method 10 times and report their mean and standard deviation (std) values. For our proposed method, we only need to implement once since no k-means is involved. The clustering performance on those four data sets is summarized in Tables 2-5, respectively. We can observe that our mPAC method achieves the best performance in most cases, which validates the effectiveness of our approach. In general, our method works better than k-means and NMF based techniques. Furthermore, it can be seen that the improvement is remarkable. With respect to graph-based clustering methods, our approach also demonstrates its superiority. In particular, both MVSC and AMGL assume that all graphs produce the same partition, while our method learns one partition for each view and finds the underlying cluster by aligning mechanism.

To visualize the effect of partitions alignment, we implement t-SNE on the clustering results of Handwritten Numerals data. As shown in Fig. 2, some partitions have a good cluster structure, thus it might be easy to find a good Y. On the other hand, although the partition of view 5 is bad, we can still achieve a good solution Y. This indicates that our method is reliable to obtain a good clustering since it operates in the partition space. By contrast, previous methods may not consistently provide a good solution.

5.3 Sensitivity Analysis

Taking Caltech7 as an example, we demonstrate the influence of parameters to clustering performance. From Fig. 3, we

[1]Our code is available: https://github.com/sckangz/mPAC

Method	F-score	Precision	Recall	NMI	ARI
KM(AllFea)	0.3834(0.0520)	0.2345(0.0463)	0.6616(0.2161)	0.1701(0.0763)	0.1561(0.0863)
Co-train	0.3094(0.0107)	0.2348(0.0034)	0.4556(0.0398)	0.1591(0.0160)	0.1144(0.0064)
Co-reg	0.3116(0.0305)	0.2337(0.0053)	0.4879(0.1173)	0.1599(0.0192)	0.1166(0.0090)
MKKM	0.3779(0.0162)	0.2359(0.0156)	0.7679(0.1402)	0.1160(0.0392)	0.1248(0.0309)
RMKM	0.3774(0.0167)	0.2476(0.0113)	0.8416(0.1563)	0.1754(0.0259)	0.1100(0.0200)
MVSC	0.3540(0.0270)	0.2459(0.0406)	0.7017(0.0801)	0.1552(0.0812)	0.1292(0.0666)
MNMF	0.3755(0.0307)	0.2685(0.0117)	**0.8558(0.1261)**	0.2576(0.0614)	0.1274(0.0515)
AMGL	0.3963(0.0167)	0.2801(0.0226)	0.6976(0.0971)	0.2686(0.0419)	0.0785(0.0399)
mPAC	**0.6780**	**0.7500**	0.6187	**0.6146**	**0.5617**

Table 2: Clustering performance on BBCSport data.

Method	F-score	Precision	Recall	NMI	ARI
KM(AllFea)	0.4688(0.0327)	0.7868(0.0080)	0.3618(0.0371)	0.4278(0.0120)	0.3172(0.0297)
Co-train	0.4678(0.0172)	0.7192(0.0136)	0.3550(0.0168)	0.3235(0.0226)	0.3342(0.0157)
Co-reg	0.4981(0.0092)	0.7014(0.0076)	0.3622(0.0098)	0.3738(0.0061)	0.2894(0.0046)
MKKM	0.4804(0.0059)	0.7659(0.0178)	0.3663(0.0040)	0.4530(0.0132)	0.3053(0.0096)
RMKM	0.4514(0.0409)	0.7491(0.0277)	0.3236(0.0376)	0.4220(0.0197)	0.2865(0.0429)
MVSC	0.3341(0.0102)	0.5387(0.0271)	0.2427(0.0130)	0.1938(0.0185)	0.1242(0.0140)
MNMF	0.4414(0.0303)	**0.7587(0.0330)**	0.3115(0.0262)	0.4111(0.0175)	0.3456(0.0576)
AMGL	0.6422(0.0139)	0.6638(0.0125)	0.6219(0.0164)	0.5711(0.0149)	0.4295(0.0208)
mPAC	**0.6763**	0.6306	**0.7292**	**0.5741**	**0.4963**

Table 3: Clustering performance on Caltech7 data.

Method	F-score	Precision	Recall	NMI	ARI
KM(AllFea)	0.3697(0.0071)	0.6235(0.0212)	0.2583(0.0095)	0.5578(0.0133)	0.2850(0.0063)
Co-train	0.3750(0.0287)	0.6375(0.0253)	0.2749(0.0238)	0.4895(0.0117)	0.3085(0.0281)
Co-reg	0.3719(0.0087)	0.6245(0.0137)	0.2882(0.0070)	0.5615(0.0042)	0.2751(0.0084)
MKKM	0.3583(0.0114)	**0.6724(0.0158)**	0.2865(0.0092)	0.5680(0.0142)	0.3039(0.0110)
RMKM	0.3955(0.0113)	0.6307(0.0144)	0.2712(0.0096)	0.5899(0.0092)	0.2952(0.0112))
MVSC	0.5417(0.0239)	0.4100(0.0245)	0.7994(0.0110)	0.4875(0.0113)	0.3800(0.0246)
MNMF	0.3643(0.0157)	0.6509(0.0119)	0.2530(0.0136)	0.5367(0.0132)	0.3128(0.0042)
AMGL	0.4017(0.0248)	0.3503(0.0479)	0.4827(0.0450)	0.5656(0.0387)	0.2618(0.0453)
mPAC	**0.5645**	0.4350	**0.8035**	**0.5986**	**0.5083**

Table 4: Clustering performance on Caltech20 data.

Method	F-score	Precision	Recall	NMI	ARI
KM(AllFea)	0.6671(0.0105)	0.6550(0.0154)	0.6889(0.0180)	0.7183(0.0106)	0.6443(0.0122)
Co-train	0.6859(0.0172)	0.6634(0.0281)	0.7109(0.0252)	0.7222(0.0149)	0.6498(0.0227)
Co-reg	0.6840(0.0269)	0.6360(0.0336)	0.6413(0.0198)	0.7583(0.0197)	0.6266(0.0314)
MKKM	0.6756(0.0000)	0.6501(0.0000)	0.7050(0.0000)	0.7526(0.0000)	0.7009(0.0000)
RMKM	0.6542(0.0258)	0.6218(0.0350)	0.6915(0.0158)	0.7431(0.0209)	0.6013(0.0300)
MVSC	0.6753(0.0335)	0.6193(0.0537)	0.7537(0.0215)	0.7566(0.0186)	0.6079(0.0419)
MNMF	0.7068(0.0272)	0.6957(0.0294)	0.7183(0.0250)	0.7431(0.0227)	0.6407(0.0056)
AMGL	0.7404(0.1070)	0.6650(0.1372)	**0.8457(0.0560)**	**0.8392(0.0543)**	0.7066(0.1235)
mPAC	**0.7473**	**0.7348**	0.7200	0.7370	**0.7069**

Table 5: Clustering performance on Handwritten numerals data.

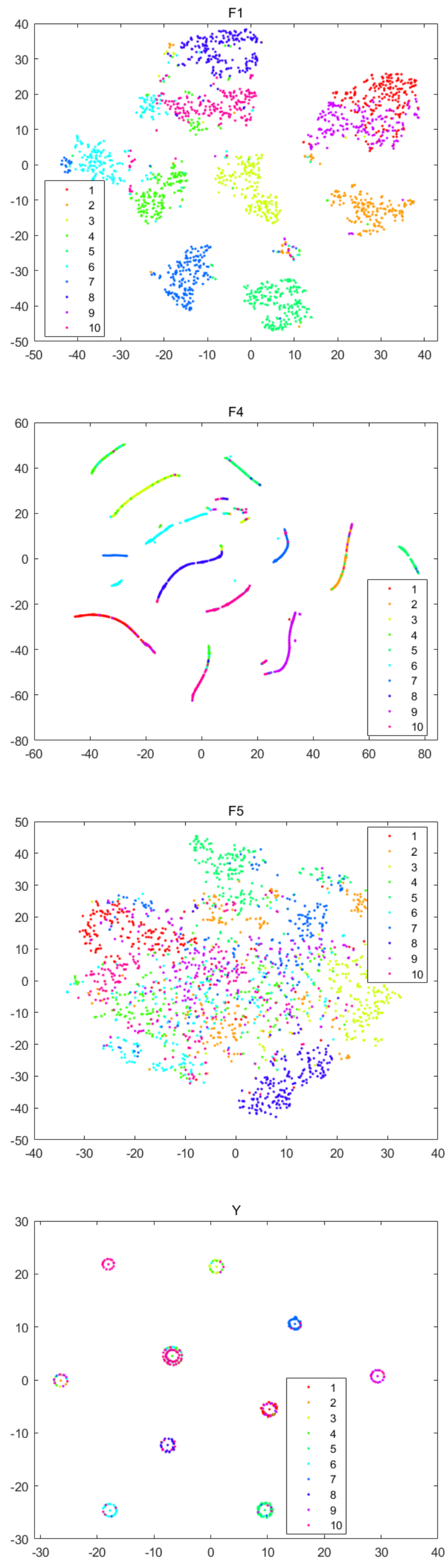
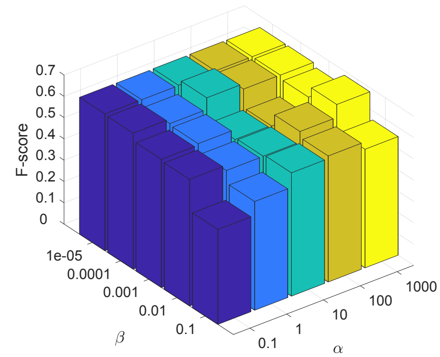
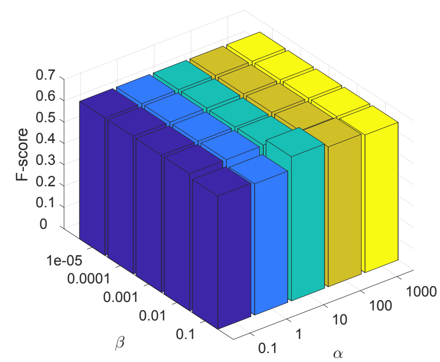


Figure 2: Some clustering results of the Handwritten Numerals data set.



(a) $\gamma = 10^{-6}$



(b) $\gamma = 10^{-3}$

Figure 3: The effect of parameters on the Caltech7 data set.

can observe that our performance is quite stable under a wide range of parameter settings. In particular, it becomes more robust to α and β when γ increases, which indicates the importance of partition alignment.

6 Conclusion

In this paper, a novel multi-view clustering method is developed. Different from existing approaches, it seeks to integrate multi-view information in partition space. We assume that each partition can be aligned to the consensus clustering through a rotation matrix. Furthermore, graph learning and clustering are performed in a unified framework, so that they can be jointly optimized. The proposed method is validated on four benchmark data sets.

Acknowledgments

This paper was in part supported by Grants from the Natural Science Foundation of China (Nos. 61806045 and 61572111), two Fundamental Research Fund for the Central Universities of China (Nos. ZYGX2017KYQD177 and A03017023701012) and a 985 Project of UESTC (No. A1098531023601041) .

References

[Cai *et al.*, 2013] Xiao Cai, Feiping Nie, and Heng Huang. Multi-view k-means clustering on big data. In *IJCAI*, pages 2598–2604, 2013.

[Chao *et al.*, 2017] Guoqing Chao, Shiliang Sun, and Jinbo Bi. A survey on multi-view clustering. *arXiv preprint arXiv:1712.06246*, 2017.

[Chen *et al.*, 2013] Xiaojun Chen, Xiaofei Xu, Yunming Ye, and Joshua Zhexue Huang. 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, 2013.

[Elhamifar and Vidal, 2013] Ehsan Elhamifar and Rene Vidal. Sparse subspace clustering: Algorithm, theory, and applications. *IEEE transactions on pattern analysis and machine intelligence*, 35(11):2765–2781, 2013.

[Gao *et al.*, 2015] Hongchang Gao, Feiping Nie, Xuelong Li, and Heng Huang. Multi-view subspace clustering. In *ICCV*, pages 4238–4246, 2015.

[Huang *et al.*, 2018] Shudong Huang, Zhao Kang, and Zenglin Xu. Self-weighted multi-view clustering with soft capped norm. *Knowledge-Based Systems*, 158:1–8, 2018.

[Huang *et al.*, 2019] Shudong Huang, Zhao Kang, Ivor W Tsang, and Zenglin Xu. Auto-weighted multi-view clustering via kernelized graph learning. *Pattern Recognition*, 88:174–184, 2019.

[Jain, 2010] Anil K Jain. Data clustering: 50 years beyond k-means. *Pattern recognition letters*, 31(8):651–666, 2010.

[Kang *et al.*, 2017] Zhao Kang, Chong Peng, and Qiang Cheng. Twin learning for similarity and clustering: A unified kernel approach. In *AAAI*, 2017.

[Kang *et al.*, 2018a] Zhao Kang, Xiao Lu, Jinfeng Yi, and Zenglin Xu. Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification. In *IJCAI*, pages 2312–2318. AAAI Press, 2018.

[Kang *et al.*, 2018b] Zhao Kang, Chong Peng, Qiang Cheng, and Zenglin Xu. Unified spectral clustering with optimal graph. In *AAAI*, 2018.

[Kang *et al.*, 2019a] Zhao Kang, Yiwei Lu, Yuanzhang Su, Changsheng Li, and Zenglin Xu. Similarity learning via kernel preserving embedding. In *AAAI*, 2019.

[Kang *et al.*, 2019b] Zhao Kang, Haiqi Pan, Steven C.H. Hoi, and Zenglin Xu. Robust graph learning from noisy data. *IEEE Transactions on Cybernetics*, pages 1–11, 2019.

[Kumar and Daumé, 2011] Abhishek Kumar and Hal Daumé. A co-training approach for multi-view spectral clustering. In *ICML*, pages 393–400, 2011.

[Kumar *et al.*, 2011] Abhishek Kumar, Piyush Rai, and Hal Daume. Co-regularized multi-view spectral clustering. In *NIPS*, pages 1413–1421, 2011.

[Liu *et al.*, 2018] Hongfu Liu, Zhiqiang Tao, and Yun Fu. Partition level constrained clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 40(10):2469–2483, Oct 2018.

[Ng *et al.*, 2002] Andrew Y Ng, Michael I Jordan, Yair Weiss, et al. On spectral clustering: Analysis and an algorithm. *NIPS*, 2:849–856, 2002.

[Nie *et al.*, 2016a] Feiping Nie, Jing Li, Xuelong Li, et al. Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification. In *IJCAI*, pages 1881–1887, 2016.

[Nie *et al.*, 2016b] Feiping Nie, Xiaoqian Wang, Michael I Jordan, and Heng Huang. The constrained laplacian rank algorithm for graph-based clustering. In *AAAI*, 2016.

[Nie *et al.*, 2018] Feiping Nie, Lai Tian, and Xuelong Li. Multiview clustering via adaptively weighted procrustes. In *SIGKDD*, pages 2022–2030. ACM, 2018.

[Peng *et al.*, 2018] Xi Peng, Canyi Lu, Zhang Yi, and Huajin Tang. Connections between nuclear-norm and frobenius-norm-based representations. *IEEE transactions on neural networks and learning systems*, 29(1):218–224, 2018.

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

[Tzortzis and Likas, 2012] Grigorios Tzortzis and Aristidis Likas. Kernel-based weighted multi-view clustering. In *ICDM*, pages 675–684. IEEE, 2012.

[Wang *et al.*, 2016] Yang Wang, Wenjie Zhang, Lin Wu, Xuemin Lin, Meng Fang, and Shirui Pan. Iterative views agreement: an iterative low-rank based structured optimization method to multi-view spectral clustering. In *IJCAI*, pages 2153–2159. AAAI Press, 2016.

[Wen and Yin, 2013] Zaiwen Wen and Wotao Yin. A feasible method for optimization with orthogonality constraints. *Mathematical Programming*, 142(1-2):397–434, 2013.

[Yang *et al.*, 2018] Xiaojun Yang, Weizhong Yu, Rong Wang, Guohao Zhang, and Feiping Nie. Fast spectral clustering learning with hierarchical bipartite graph for large-scale data. *Pattern Recognition Letters*, 2018.

[Zhan *et al.*, 2017] Kun Zhan, Changqing Zhang, Junpeng Guan, and Junsheng Wang. Graph learning for multiview clustering. *IEEE transactions on cybernetics*, (99):1–9, 2017.

[Zhang *et al.*, 2017] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. Latent multi-view subspace clustering. In *CVPR*, pages 4279–4287, 2017.

[Zong *et al.*, 2017] Linlin Zong, Xianchao Zhang, Long Zhao, Hong Yu, and Qianli Zhao. Multi-view clustering via multi-manifold regularized non-negative matrix factorization. *Neural Networks*, 88:74–89, 2017.