Multi-view Subspace Clustering via Partition Fusion

Juncheng Lv[a], Zhao Kang[a], Boyu Wang[b], Luping Ji[a], Zenglin Xu[a]

[a]*School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, China*
[b]*Department of Computer Science, University of WesternOntario, Canada*

Abstract

Multi-view clustering is an important approach to analyze multi-view data in an unsupervised way. Among various methods, the multi-view subspace clustering approach has gained increasing attention due to its encouraging performance. Basically, it integrates multi-view information into graphs, which are then fed into spectral clustering algorithm for final result. However, its performance may degrade due to noises existing in each individual view or inconsistency between heterogeneous features. Orthogonal to current work, we propose to fuse multi-view information in a partition space, which enhances the robustness of Multi-view clustering. Specifically, we generate multiple partitions and integrate them to find the shared partition. The proposed model unifies graph learning, generation of basic partitions, and view weight learning. These three components co-evolve towards better quality outputs. We have conducted comprehensive experiments on benchmark datasets and our empirical results verify the effectiveness and robustness of our approach.

Keywords: `Multi-view learning`, `subspace clustering`, `Partition space`, `Information fusion`

1. Introduction

In many real-world problems, data are collected from different sources in diverse domains or described by various feature collectors [1, 2, 3, 4, 5]. For in-

---

stance, human activities can be captured by RGB video cameras, depth cameras, or on-body sensors [6, 7]; pictures shared on websites are usually surrounded by textual tags and descriptions; the same news is reported in various countries with different languages. To process these kinds of data, a number of multi-view learning algorithms have been developed [8, 9, 10, 11, 12]. These methods are based on two fundamental assumptions: consistency and complementarity across views. That is to say, there exists consistent information which is shared by all views and complementary knowledge that is not contained in all views. Therefore, it is crucial for learning algorithms to make good use of the multi-view information. For example, instead of relying on a single view, multi-view clustering, attempts to achieve a better performance by integrating compatible and complementary information from different views [13, 14, 15, 16].

In recent years, plenty of multi-view clustering techniques have been proposed [17, 18, 19, 20, 21, 22]. They can be roughly divided into the following categories. First, the co-training style algorithms intend to maximize the mutual agreement across all views and arrive at their broadest consensus [23, 24, 25]. Second, kernel-based methods use pre-defined kernels corresponding to different views and then combine these kernels either linearly or non-linearly in order to improve the clustering performance[26, 27, 28, 29]. Third, graph-based methods have been derived from traditional spectral clustering with the help of some similarity measures[30, 31, 32, 12]. Fourth, subspace clustering based methods. Subspace clustering tries to find underlying subspaces such that all data points can be segmented correctly and each group fits into one of the low-dimensional subspaces [33, 34, 35, 36, 37, 38, 39]. Due to its promising performance, varieties of multi-view subspace clustering methods have been proposed in the last few years [40, 41, 42].

Most multi-view subspace clustering methods learn the sample affinity graph matrix of each view by deploying features of different views, and then build a consensus graph S [40, 41, 43]. Some other approaches directly learn a common graph matrix S [44]. Subsequently, spectral clustering algorithm [45, 46] is implemented on the graph Laplacian constructed by S to obtain the final

clustering result. Therefore, these methods ahopt a two-steps procedure. Since data are often noisy or corrupted in practice, one critical issue with these approaches is that the learned graph is often noisy, inaccurate, and fails to reveal the true relationships between data points [47, 48]. In return, this fixed graph will deteriorate the downsteam clustering task, which makes the entire learning procedure suboptimal.

Orthogonally to integrating the multi-view information into a single graph, in this paper, we propose to fuse partitions to improve the model robustness. Note that the underlying assumption for multi-view clustering is that there exists a unique clustering pattern shared by all views. Hence, the partition space should be more robust to noise. On the one hand, even if the graphs are contaminated, the cluster structure might be slightly influenced or even remain intact. On the other hand, even if some of the partitions are severely damaged, one could still obtain a reasonable performance based on our partition fusion technique. In specific, we adaptively learn a weight for each view to control its contribution to the final clustering. The final clustering is achieved through a purposely designed weighting mechanism imposed on basic partitions.

In summary, the main contributions of this work are three-fold:

1. We propose to fuse multi-view information in a partition space. A novel fusion mechanism is further developed to find the consensus clustering.

2. We present a unified multi-view subspace clustering model which iteratively learns a graph for each view, a partition for each view, a weight for each pratition, and a consensus clustering. By leveraging the inherent interactions between these four subtasks, they enhance each other.

3. Extensive experiments on benchmark data sets validate the effectiveness of our model. Experiments on noisy data demonstrate the robustness of our approach.

The rest of this paper is organized as follows. After a review of the related work in Section 2, we introduce our proposed framework for multi-view subspace learning in Section 3. In Section 4, we evaluate the clustering performance of

our algorithm. The robustness evaluation of the proposed framework on noisy data sets is presented in Section 5. Finally, we conclude the paper in Section 6.

Notation Summary In this paper, we represent the matrix with capital letter and vector with lower case letter. For a matrix $A \in \mathbb{R}^{m \times n}$, $A_{i,:}$ and $A_{:,j}$ represent the i-th row and j-th column of A, respectively. The ℓ_2-norm of vector x is denoted by $\|x\|_2 = \sqrt{x^\top \cdot x}$, where $\top$ stands for the transpose operation. The trace operator is written as $Tr(\cdot)$. The definition of Frobenius norm of A is $\|A\|_F = \sqrt{\sum_{i=1}^m \sum_{j=1}^n A_{ij}^2}$. $A \geq 0$ indicates all elements of A are nonnegative. I is the identity matrix with a proper size.

2. Related Work

Multi-view Subspace Clustering (MVSC) is developed on the basis of subspace clustering. We denote the multi-view data X with $[X^1, X^2, \cdots, X^t] \in \mathbb{R}^{m \times n}$, where $X^v \in \mathbb{R}^{m_v \times n}$ represents the v-th view data with m_v features and $m = \sum_v m_v$. Basically, [40, 49] propose to learn a graph on individual view by solving

$$\min_{S^v} \sum_{v=1}^t \|X^v - X^v S^v\|_F^2 + \alpha f(S^v) \ \ s.t. \ S^v \geq 0, \tag{1}$$

where $S^v \in \mathbb{R}^{n \times n}$ denotes the graph for the v-th view, $f(\cdot)$ represents a certain regularizer function, and α is a regularization parameter to balance the model complexity and the fitting loss. S^v is also called the self-expression coefficient matrix, which expresses each sample as a linear combination of other samples. Hence, it can measure the similarities between samples. Based on it, subspace clustering implements spectral clustering algorithm [45].

In [40], the authors enforce that all graphs share a unique cluster indicator matrix, i.e.,

$$\min_F \sum_v Tr(F^\top L^v F) \ \ s.t. \ \ F^\top F = I, \tag{2}$$

where graph Laplacian $L^v = D^v - S^v$ with diagonal matrix D^v defined as $d_{ii}^v = \sum_j s_{ij}^v$, $F \in \mathbb{R}^{n \times c}$ is the cluster indicator matrix, and c is number of

4

clusters. It is obvious that F is negotiated across different graphs, i.e., different L^v's correspond to the unique F. Therefore, the final F might not be optimal.

Different from the above approach, the authors in [43, 49] first calculate the consensus graph S based on averaging, i.e., $\sum_{v=1}^{t} S^v/t$. Then, spectral clustering is applied. This simple approach fails to distinguish the different contributions of various views. Afterwards, we can obtain the final cluster indicators by performing k-means on F.

Another class of methods just consider the consistent graph of all views and their objective function can be written as [44, 50]

$$\min_S \sum_{v=1}^{t} \|X^v - X^v S\|_F^2 + \alpha \mathcal{R}(S) \ s.t. \ S \geq 0, \tag{3}$$

where $\mathcal{R}(Z)$ is some regularization function which varies in different algorithms. For instance, researchers in [44] use popular low-rank and sparse regularizers simultaneously. It is easy to see that just one graph can not preserve the flexible local mannifold structure of different views [43].

We can observe that above approaches integrate multi-view information in the data space and their performance totally depends on the quality of graph. Once the graph is fixed, the rest of MVSC process reduces to spectral clustering, which is not subject to change. In real-world applications, data are often contaminated due to noise or outliers, which deteriorates the resulting graph. Consequently, the clustering performance will degrade. Therefore, we argue that direct manipulation on graphs might not be a good idea to make use of multi-view information.

Instead, we propose to integrate multi-view knowledge by fusing partitions. Concretely, we generate one partition for each view, which forms the basic partitions. Then, we seek for a consensus clustering from them based on our purposely designed weighting mechanism. This change in operation domain accompanies a number of advantages. First, each partition will capture the intrinsic cluster structure, so it is easy to find an agreement among all partitions. Second, even if one partition is heavily distorted, its contribution can be reduced

by assigning a small weight, so as to prevent it from adversely affecting consensus clustering. As a result, this approach could be robust to noise.

Note that the proposed approach is different from ensemble clustering [51, 52] in several aspects. First, only one partition is generated for each view. On the contrary, many partitions are often produced for ensemble clustering. Second, the basic partitions are integrated differently. For example, Tao et al.[52] adopt a low-rank and sparse decomposition strategy to discover the connection among views and detect noises. Third, a unified framework is utilized, i.e., the generation of basic partitions and their integration are joint performed. In contrast, basic partitions are input of ensemble clustering.

3. Proposed Methodology

In this section, we will introduce our novel model and its solution.

3.1. Formulation

As in aforementioned MVSC [40, 53], we combine graph construction and spectral clustering together. Different from it, we generate one partition for each view. Multiple partitions allow us to manipulate multi-view information in a partition space, which enhances the model robustness. In specific, we have

$$
\min_{S^v, F_v} \sum_{v=1}^{t} \|X^v - X^v S^v\|_F^2 + \alpha Tr(F_v^\top L^v F_v) + \beta \|S^v\|_F^2
$$

$$
s.t. \quad F_v^\top F_v = I, \ S^v \geq 0,
$$

(4)

where $F_v \in \mathbb{R}^{n \times c}$ is the partition result for view v. We can see that Eq. (4) will provides multiples partitions. For multi-view clustering, all views share a unique cluster pattern. Due to noises in defferent views or the heterogeneity of features, these partition matrices are not identical in general. Therefore, our next goal is to find a consensus clustering from them.

To treat each view discriminatively, we propose a weighting mechanism. To this end, we introduce a variable w_v for view v which characterizes the

importance of view v. Larger value means more contribution of this partition to consensus clustering. It is reasonable to propose the following objective function

$$\min_{w_v, S^v, F_v} \sum_{v=1}^{t} w_v \Big\{ \|X^v - X^v S^v\|_F^2 + \alpha Tr(F_v^\top L^v F_v) + \beta \|S^v\|_F^2 \Big\}$$

$$s.t. \sum_v w_v = 1, \ w_v \geq 0, \ F_v^\top F_v = I, \ S^v \geq 0 \tag{5}$$

If the loss in the bracket is small, weight w_v will have a large value; vice versa.

After we obtain the weight for each view, how can we reach the consensus clustering, i.e., a partition that fits all views? Unlike classification or regression, the cluster indicator matrix for each view is not unique. In general, for each unique clustering with c clusters, there are $c!$ (c factorial) equivalent representations [54]. So the popular Euclidean distance is not applicable to measure the distance between indicator matrices any more and we need to figure out how to define the distances between partitions.

To address above challenge, we propose to use inner product $F_v F_v^\top$, which actually represents the similarities among all data points in the v-th view. It is easy to understand that it is invariant with respect to $c!$ permutations. Hence, we can measure the partition distance in terms of $F_v F_v^\top$. Specifically, we develop the following partition fusion objective function

$$\min_{Y \in \mathbb{R}^{n \times c}, Y^\top Y = I} \|YY^\top - \sum_{v=1}^{t} w_v F_v F_v^\top\|_F^2, \tag{6}$$

where Y is the consensus cluster indicator matrix. If w_v has a large value, the v-th partition will contribute a lot to final Y.

Finally, we can combine Eqs. (5) and (6) to form a unified objective function, so that each variable can be iteratively updated. Our proposed Partition Fusion

multi-view Subspace Clustering (PFSC) can be formulated as

$$\min_{S^v, F_v, w_v, Y} \sum_{v=1}^{t} w_v \Big\{ \|X^v - X^v S^v\|_F^2 + \alpha Tr(F_v^\top L^v F_v) + \beta \|S^v\|_F^2 \Big\}$$

$$+ \|YY^\top - \sum_{v=1}^{t} w_v F_v F_v^\top\|_F^2 \qquad (7)$$

$$s.t. \sum_{v=1}^{t} w_v = 1, \, w_v \geq 0, \, S^v \geq 0$$

$$F_v^\top F_v = I, \, Y^\top Y = I.$$

With input data X, PFSC will output cluster indicator matrix Y. Hence, it is an end-to-end clustering method. Compared to existing work in the literature, it enjoys the following properties.

- Orthogonal to existing multi-view clustering methods, our proposed model integrates multi-view information in a partition space. Since all partitions admit a unique cluster pattern, it is natural to implement information fusion in a partition space.

- A weight is dynamically learned for each view, which can identify the importance of each view.

- This unified framework seamlessly integrates the graph learning, spectral clustering, weight learning, and partition fusion. By iteratively updating S, F, w, Y, they can be repeatly improved. This joint learning strategy facilitates to obtain a better solution.

3.2. Optimization

Since Eq.(7) involves several coupled variables, it is difficult to solve it. Therefore, we divide the original problem into four subproblems and develop an alternating and iterative algorithm. S^v, F_v, w_v, Y can be solved effectively and individually by fixing the others.

S^v**-subproblem**: By fixing F_v, w_v, and Y, we update S^v by solving:

$$\min_{S^v} \sum_v \Big\{ \|X^v - X^v S^v\|_F^2 + \alpha Tr(F_v^\top L^v F_v) + \beta \|S^v\|_F^2 \Big\}. \qquad (8)$$

Note that S^v's are independent for each view, hence we can solve them separately. For convenience, we ignore the subscript/superscript tentatively and get

$$\min_S Tr(-2X^\top XS + S^\top X^\top XS) + \alpha Tr(F^\top LF) + \beta Tr(S^\top S). \qquad (9)$$

Setting the the derivative of Eq. (9) with respect to S to zero, and noting that $\sum_{i,j} \frac{1}{2}\|F_{i,:} - F_{:,j}\|_2^2 s_{ij} = Tr(F^\top LF)$, we obtain the analytical solution for $S_{:,i}$,

$$S_{:,i} = (X^\top X + \beta I)^{-1}(X^\top X_{:,i} - \frac{\alpha}{4}d_i), \qquad (10)$$

where $d_i \in \mathbb{R}^{n \times 1}$ is a vector with the j-th element as $d_{ij} = \|F_{i,:} - F_{j,:}\|_2^2$. Note that once parameter α is given, the inverse is fixed in each iteration. In other words, we can calculate it in advance to save computing time. After obtaining S, we can set its negative elements to zero to ensure its nonegative.

F_v-**subproblem**: Omitting all other unrelated terms with respect to F_v, we obtain

$$\min_{F_v, F_v^\top F_v = I} \alpha \sum_v w_v Tr(F_v^\top L^v F_v) + \|YY^\top - \sum_v w_v F_v F_v^\top\|_F^2. \qquad (11)$$

Similarly, F_v can be solved separately for each view. Then, we have:

$$\min_{F_v, F_v^\top F_v = I} Tr(F_v^\top M F_v), \qquad (12)$$

where $M = \alpha L_v + w_v I - 2YY^\top - 2\sum_{j \neq v} F_j F_j^\top$. Then, the optimal solution F_v is the c eigenvectors of M corresponding to the c smallest eigenvalues.

w_v-**subproblem**: Our problem can be simplified as

$$\min w^\top Pw - qw$$
$$\text{s.t.} \quad \sum_v w_v = 1, \quad w_v \geq 0, \qquad (13)$$

where $P \in \mathbb{R}^{n \times n}$ with $P_{ij} = Tr[F_i F_i^\top \times F_j F_j^\top]$ and q is a vector with

$$q_i = -g_i + 2Tr(YY^\top F_i F_i^\top),$$
$$g_i = \|X^i - X^i S^i\|_F^2 + \alpha Tr(F_i^\top L^i F_i) + \beta \|S^i\|_F^2. \qquad (14)$$

It is a standard quadratic programming problem, which can be solved efficiently.

Y-**subproblem**: As for Y, we have the following equivalent formulation

$$\min_{Y,Y^\top Y=I} Tr[Y^\top (I - 2\sum_v w_v F_v F_v^\top)Y], \tag{15}$$

which can be solved by singular value decomposition (SVD).

The entire optimization procedure for (7) is summarized in **Algorithm 1**[1]. The main computational burden lies in the matrix inversion (line 2) and SVD (line 6). Since we just consider the top c eigenvectors, the complexity of SVD is $O(n^2 c)$. Therefore, the overall time complexity of Algorithm 1 is $O(n^3)$.

Algorithm 1: Optimization for PFSC

Input: multi-view data matrix $X^1, \cdots, X^t$, cluster number c, parameters α, β.

Output: S^v, F_v, w_v, Y.

Initialize: Random matrix F_v, $w_v = 1/t$.

REPEAT

1: **for** view 1 to v **do**

2: Update each column of S according to (10);

3: S=max(S,0);

4: Solve the subproblem (12);

5: Update w_v via (13).

6: **end for**

7: Solve the subproblem (15).

UNTIL stopping criterion is met.

---

4. Experiments on Benchmark Data

4.1. Data sets

To fully assess the effectiveness of our proposed method, we conduct experiments on three widely used data sets with six types of features.

---

[1] Our source code will be released later.

Table 1: Description of the datasets. The feature number for each view is shown in parenthesis.

View	HW	Caltech7	Caltech20
1	Profile correlations (216)	Gabor(48)	Gabor (48)
2	Fourier coefficients (76)	Wavelet moments (40)	Wavelet moments (40)
3	Karhunen coefficients (64)	CENTRIST (254)	CENTRIST (254)
4	Morphological (6)	HOG (1984)	HOG (1984)
5	Pixel averages (240)	GIST (512)	GIST (512)
6	Zernike moments (47)	LBP (928)	LBP (928)
Data points	2000	1474	2386
Classes	10	7	20

Handwritten numerals (HW) [2] data set is selected from UCI machine learning repository. It is comprised of 2000 data points for 0 to 9 digit classes, and each class has 200 data points.

Caltech7 [3] is an object recognition data set with 7 categories, including "Face, Snoopy, Garfield, Motorbikes, Dolla-Bills, Stop-sign, and Windsor-chairs".

Caltech20 contains 20 classes "Brain, Camera, Face, Ferry, Rhino, Pagoda, Snoopy, Wrench, Stapler, Leopards, Hedgehog, Garfield, Binocular, Motorbikes, Windsor Chair, Car-Side, Dolla-Bill, Stop-Sign, Yin-yang, and Water-Lilly".

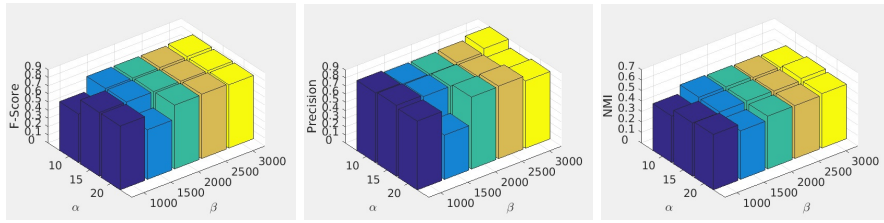The specific characteristics of these data sets are summarized in Table 1.

Figure 1: Sensitivity analysis of parameters for our method over Caltech7 dataset evaluated with F-Score, Precision, and NMI.

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

- The classic k-means clustering algorithm (KM): The KM clustering on concatenated features is used as a baseline algorithm.

- Co-regularized multi-view spectral clustering (Co-reg) [25]: A co-regularization mechnism is utilized to ensure that partitions from different views are close to each other.

- Co-trained multi-view spectral clustering (Co-train) [24]: A co-training approach is used to learn multiple Laplacian eigenspace.

- Multi-view kernel k-means clustering (MKKM) [29]: In MKKM, data are first mapped into high-dimensional space by kernel trick. Then, kernels from different views are combined based on a weighting principle.

- Robust multi-view k-means clustering (RMKM) [55]: To cope with outliers, this method adopts structured sparsity-inducing norm to integrate multi-view information.

- Multi-view subspace clustering (MVSC) [40]: This method simultaneously learns multiple graphs and forces them generate the same cluster structure.

- Multi-manifold regularized nonnegative matrix factorization (MNMF) [56]: Based on NMF, this method preserves the local geometrical structure of multi-view data.

- Auto-weighted multiple graph learning (AMGL) [32]: Different from Eq. (2), a weight is assigned for each view.

- Multi-View Ensemble Clustering (MVEC) [52]: Solve multi-view clustering in an ensemble clustering way.

- Multiple Partitions Aligned Clustering (mPAC) [57]: This recent method uses a different approach to combine basic partitions.

Table 2: Clustering performance on Caltech7 data.

Method	F-Score	Precision	Recall	NMI	Adj-RI
KM	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)
MVEC	0.5862(0.0244)	0.4523(0.0114)	**0.8363(0.0654)**	**0.5924(0.0285)**	0.4318(0.0409)
mPAC	**0.6763**	0.6306	**0.7292**	**0.5741**	**0.4963**
PFSC	**0.7627(0.0117)**	**0.8687(0.0656)**	0.6836(0.0280)	0.4388(0.0119)	**0.5832(0.0035)**

Following [55], we pre-process the data based on normalization, i.e., all values range from -1 to 1. For a fair comparison, we adopt five popular evaluation metrics as shown in [24]: F-Score, Precision, Recall, normalized mutual information (NMI) and Adjusted Rand Index (Adj-RI). For all of them, a larger value indicates a better clustering performance. For all methods, we either adopt the default parameter values given by the respective authors or tune them to achieve the best performance. Each method is repeated 10 times and the mean and standard deviation (std) values are reported.

4.3. Experimental Result

Tables 2-4 show the clustering results on the three benchmark data sets. For each measure, the best two methods are highlighted in boldface. In most cases, our proposed method achieves the best clustering performance in comparison with other state-of-the-art multi-view clustering methods. In particular,

1. Our proposed PFSC often performs much better than current MVSC algorithm. Recall that both MVSC and PFSC learn multiple graphs. MVSC forces each graph to generate the same cluster, which is too restrictive

Table 3: Clustering performance on Caltech20 data.

Method	F-Score	Precision	Recall	NMI	Adj-RI
KM	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)
MVEC	0.5229(0.0377)	0.4366(0.0412)	0.7187(0.0499)	0.5841(0.0114)	**0.4517(0.0416)**
mPAC	**0.5645**	0.4350	**0.8035**	**0.5986**	**0.5083**
PFSC	0.5163(0.0396)	0.4337(0.0492)	0.6432(0.0402)	0.5790(0.0178)	0.4437(0.0415)

considering the heterogeneous nature of views. PFSC produces a partition for each graph and fuses them to reach a consensus which allows for greater flexibility to deal with heterogeneous data.

2. With respect to AMGL, another representative spectral clustering based method, PFSC outperforms it considerably in most cases. Though it assigns a weight for each graph, a common clustering is still assumed for all graphs. Moreover, its weight is empirically determined, which might not be correct for complex data. These factors make AMGL impossible to perform well in all cases.

3. Compared to the multi-view ensemble clustering method MVEC, our method PFSC performs better in most cases. As we mention previously, they are different in several key aspects that are responsible for our improvements.

4. In general, graph-based methods perform better than k-means and NMF approaches, which is due to the efficiency of graph representation for the data in a non-Euclidean space [58]. Graphs also provide a convenient way to explore supplementary information from multiple views.

5. With respect to mPAC, our method gives competitive performance. In

Table 4: Clustering performance on Handwritten numerals data.

Method	F-Score	Precision	Recall	NMI	Adj-RI
KM	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)**
MVEC	0.7196(0.0313)	**0.8082(0.0157)**	0.6501(0.0437)	**0.8166(0.0142)**	0.6847(0.0361)
mPAC	**0.7473**	0.7348	0.7200	0.7370	**0.7069**
PFSC	0.7263(0.0249)	**0.7549(0.0176)**	0.7001(0.0326)	0.7666(0.0148)	0.6948(0.0281)

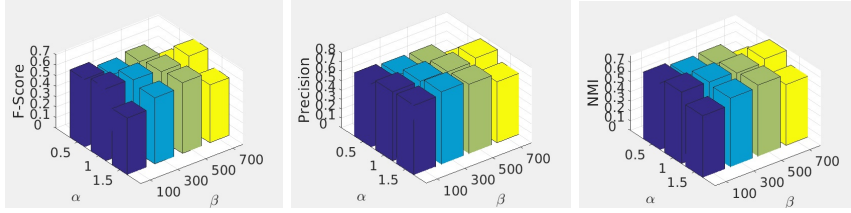some cases, our performance is even better.

Figure 2: Sensitivity analysis of parameters for our method over Caltech20 dataset evaluated with F-Score, Precision, and NMI.

To sum up, PFSC obtains competitive performance with respect to state-of-the-art techniques. This validates the effectiveness of partition fusion strategy.

4.4. Parameter Sensitivity Analysis

There are two tuning parameters, α and β in our model (7). Taking Caltech20 and Caltech7 as examples, we empirically examine their influence on F-score, Precision, and NMI, as shown in Figures 1 and 2. It can be observed that the performance of PFSC is relatively stable for a wide range of parameters.

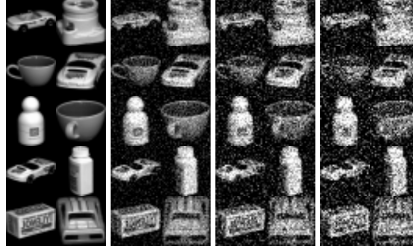## 5. Experiments on Noisy Data

Figure 3: The first column is some COIL20 images. They are corrupted by Gaussian noise with different variance: 0.01, 0.03, 0.05 (from 2nd to 4th column).
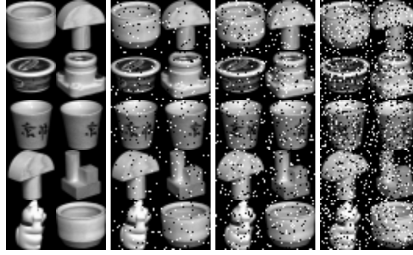

Figure 4: The first column is some COIL20 images. They are corrupted by Salt & Pepper noise with different density: 0.05, 0.1, 0.2 (from 2nd to 4th column).

In practice, images are usually liable to suffer from noises. To examine the robustness of our model, we contaminate clean images by adding two commonly seen noises: Gaussian noise and Salt & Pepper noise. We first construct a single-view data set by randomly sampling 24 images for each class from COIL20 [59] which is a data set of 20 toy images. This results in a data set with size 1024×480. Then, we add three levels of noise to obtain three noise data. Based on them, we form a three-view data set. In specific, we corrupt the images with Gaussian noise with zero mean and variance 0.01, 0.03, and 0.05; Salt & Pepper noise with density 0.05, 0.1, and 0.2. Some sample images are displayed in Figures 3 and 4. We can observe that those images are heavily corrupted.

Here, we run k-means on all partitions from our model (7), i.e., F_1, F_2, F_3, Y, and report their results as PFSC-F_1, PFSC-F_2, PFSC-F_3, PFSC-Y,

Table 5: Clustering performance on noisy data.

Noise	Method	F-Score	Precision	Recall	NMI	Adj-RI
Gaussian	MVSC	0.3467(0.0321)	0.5194(0.0195)	0.2294(0.0347)	0.5615(0.0229)	0.2677(0.0373)
	AMGL	0.4240(0.0734)	0.3076(0.0696)	**0.7036(0.0340)**	**0.7754(0.0343)**	0.3822(0.0820)
	PFSC-F_1	0.5156(0.0528)	0.6439(0.0366)	0.4469(0.0629)	0.7506(0.0239)	0.4908(0.0576)
	PFSC-F_2	0.5200(0.0558)	0.6595(0.0364)	0.4312(0.0632)	0.7525(0.0279)	0.4902(0.0605)
	PFSC-F_3	0.5086(0.0267)	**0.6613(0.0232)**	0.4143(0.0338)	0.7483(0.0163)	0.4777(0.0293)
	PFSC-Y	**0.5226(0.0395)**	0.6257(0.0216)	0.4508(0.0529)	0.7590(0.0200)	**0.4943(0.0431)**
Salt & Pepper	MVSC	0.3510(0.0363)	0.5034(0.0240)	0.2710(0.0397)	0.5709(0.028)	0.3076(0.0408)
	AMGL	0.3954(0.0464)	0.2800(0.0451)	**0.6836(0.0189)**	0.7539(0.0175)	0.3509(0.0520)
	PFSC-F_1	0.5065(0.0459)	0.6042(0.0336)	0.4290(0.0566)	0.7493(0.0230)	0.4765(0.0502)
	PFSC-F_2	0.5218(0.0583)	0.6105(0.0489)	0.4575(0.0651)	0.7516(0.0291)	0.4939(0.0629)
	PFSC-F_3	0.5016(0.0398)	0.6118(0.0393)	0.4284(0.0539)	0.7443(0.0212)	0.4717(0.0436)
	PFSC-Y	**0.5325(0.0345)**	**0.6190(0.0222)**	0.4683(0.0429)	**0.7593(0.0175)**	**0.5055(0.0373)**

respectively. We compare PFSC with two most relevant gragh-based multi-view clustering method: MVSC and AMGL. In addition, we choose them since they produce very competitive performance in Tables 3-4. In fact, MVSC is a robust learning algorithm, in which an error variable E^v is purposely introduced to characterize noise for each view, and then ℓ_1-norm regularization is imposed to E^v to enforce the sparsity for the outlying entries.

Table 5 summarizes the learning performances of the algorithms on noisy data, and it can be observed that PFSC-Y outperforms MVSC and AMGL significantly by more than 10% in terms of F-Score, Precision, and Adj-RI. Though MVSC is designed to resist noise, it is inferior to PFSC-Y in all cases. This is due to the fact that MVSC treats each view equally and assumes that all graphs share the same partition matrix. To reach the overall minimum of objective function, the final cluster pattern could be deteriorated. On the other
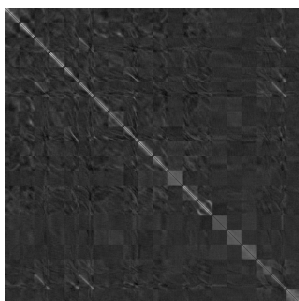
(a) The 1st View
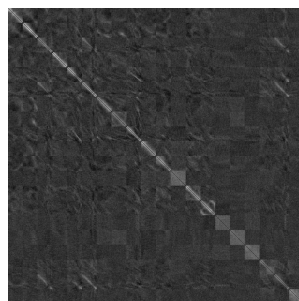
(b) The 2nd View
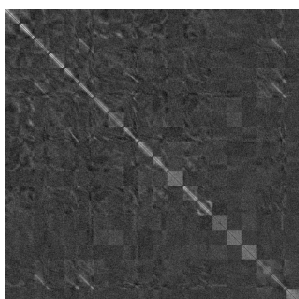
(c) The 3rd View

Figure 5: The learned graphs S corrupted by Gaussian noise.

(a) Partition F_1 (b) Partition F_2

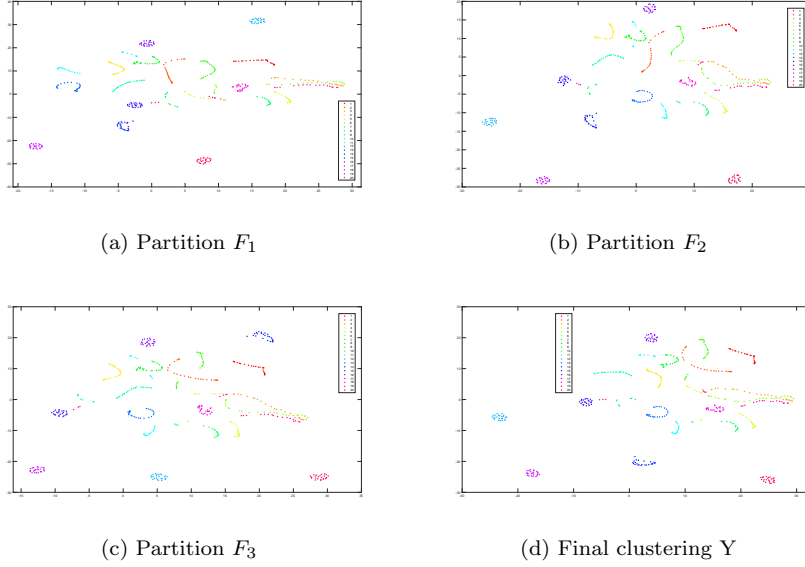(c) Partition F_3 (d) Final clustering Y

Figure 6: Visualization of partitions corrupted by Gaussian noise.

hand, both AMGL and PFSC-Y adopt a weighting strategy, hence they can assign more weight to high quality view. Different from AMGL, which uses the same partition for all views, PFSC-Y finds the final clustering through fusing multiple basic partitions. Since all views admit the same cluster pattern, it becomes easy for PFSC to search for the best one.

Furthermore, we can observe that our final clustering PFSC-Y often outperforms PFSC-F_1, PFSC-F_2, and PFSC-F_3. This validates the effectiveness of our fusion strategy. Based on those partitions from each single view, we can eventually achieve a better clustering. The learned graphs and partitions are also displayed in Figures 5-8. To obtain Figures 6 and 8, t-SNE is implemented [60]. We can see that the learned graphs can not present the cluster structure. Ideally, they should be block-diagonal. On the other hand, the partitions can explicitly display the cluster structure. This demonstrates the robustness of our model.

19
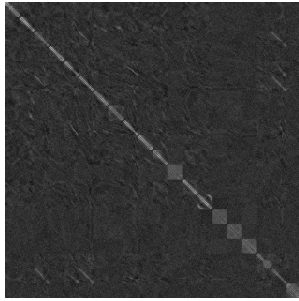
(a) The 1st View
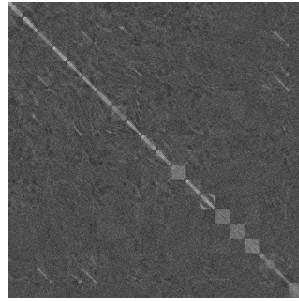
(b) The 2nd View
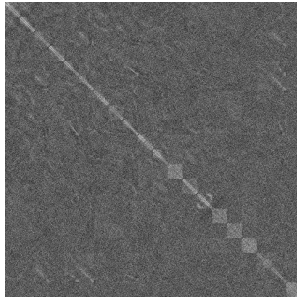
(c) The 3rd View

Figure 7: The learned graphs S corrupted by Salt & Pepper noise.

(a) Partition F1   (b) Partition F2

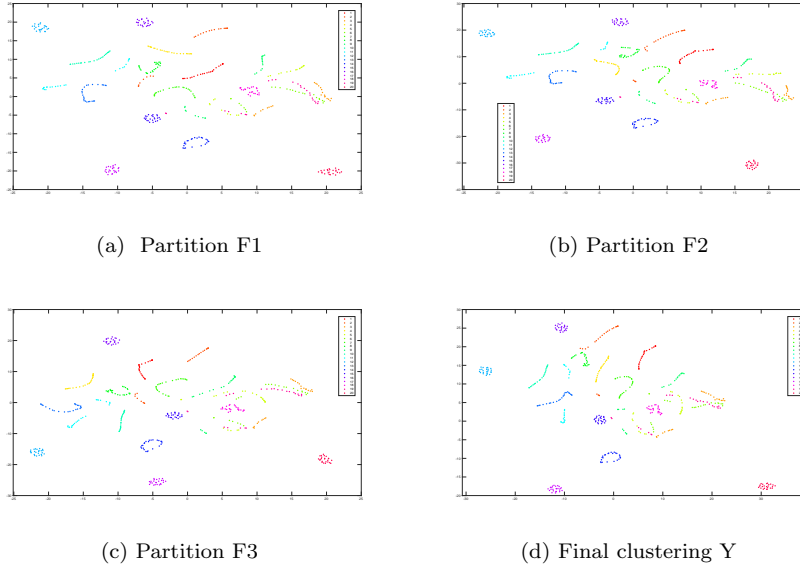(c) Partition F3   (d) Final clustering Y

Figure 8: Visualization of partitions corrupted by Salt & Pepper noise.

6. Conclusion

In this paper, we develop a novel multi-view subspace clustering method, aiming to exploring multi-view information in a partition space to enhance model robustness. The proposed model integrates graph learning, spectral clustering, weight learning, and partition fusion into a unified framework, in which each component is optimized. Experimental results on widely adopted benchmark data sets validate the effectiveness and robustness of the proposed method.

References

[1] Z. Ding, M. Shao, Y. Fu, Robust multi-view representation: A unified perspective from multi-view learning to domain adaption., in: IJCAI, 2018, pp. 5434–5440.

[2] M. Yin, J. Gao, S. Xie, Y. Guo, Multiview subspace clustering via tenso-

rial t-product representation, IEEE Transactions on Neural Networks and Learning Systems 30 (3) (2018) 851–864.

[3] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 1582–1590.

[4] Z. Zhang, Y. Zhang, G. Liu, J. Tang, S. Yan, M. Wang, Joint label prediction based semi-supervised adaptive concept factorization for robust data representation, IEEE Transactions on Knowledge and Data Engineering.

[5] Y. Wang, L. Wu, X. Lin, J. Gao, Multiview spectral clustering via structured low-rank matrix factorization, IEEE transactions on neural networks and learning systems 29 (10) (2018) 4833–4843.

[6] S. Li, Y. Li, Y. Fu, Multi-view time series classification: A discriminative bilinear projection approach, in: Proceedings of the 25th ACM International on Conference on Information and Knowledge Management, ACM, 2016, pp. 989–998.

[7] S. Yao, G. Yu, J. Wang, C. Domeniconi, X. Zhang, Multi-view multiple clustering, in: IJCAI, 2019.

[8] C. Xu, D. Tao, C. Xu, A survey on multi-view learning, arXiv preprint arXiv:1304.5634.

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

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

[11] Z. Hu, F. Nie, R. Wang, X. Li, Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding, Information Fusion 55 (2020) 251–259.

[12] K. Zhan, F. Nie, J. Wang, Y. Yang, Multiview consensus graph clustering, IEEE Transactions on Image Processing 28 (3) (2018) 1261–1270.

[13] S. Huang, H. Wang, D. Li, Y. Yang, T. Li, Spectral co-clustering ensemble, Knowledge-Based Systems 84 (2015) 46–55.

[14] M. B. Blaschko, C. H. Lampert, Correlational spectral clustering, in: 2008 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2008, pp. 1–8.

[15] K. Chaudhuri, S. M. Kakade, K. Livescu, K. Sridharan, Multi-view clustering via canonical correlation analysis, in: Proceedings of the 26th annual international conference on machine learning, ACM, 2009, pp. 129–136.

[16] Y. Wang, X. Lin, L. Wu, W. Zhang, Q. Zhang, X. Huang, Robust subspace clustering for multi-view data by exploiting correlation consensus, IEEE Transactions on Image Processing 24 (11) (2015) 3939–3949.

[17] G. Chao, S. Sun, J. Bi, A survey on multi-view clustering, arXiv preprint arXiv:1712.06246.

[18] K. Zhan, C. Zhang, J. Guan, J. Wang, Graph learning for multiview clustering, IEEE transactions on cybernetics (99) (2017) 1–9.

[19] X. Liu, X. Zhu, M. Li, C. Tang, E. Zhu, J. Yin, W. Gao, Efficient and effective incomplete multi-view clustering, in: Proc. AAAI, 2019, pp. 1–8.

[20] W. Zhuge, C. Hou, X. Liu, H. Tao, D. Yi, Simultaneous representation learning and clustering for incomplete multi-view data, in: Proceedings of the 28th International Joint Conference on Artificial Intelligence, AAAI Press, 2019, pp. 4482–4488.

[21] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, W. Gao, Late fusion incomplete multi-view clustering, IEEE transactions on pattern analysis and machine intelligence.

[22] C.-D. Wang, J.-H. Lai, S. Y. Philip, Multi-view clustering based on belief propagation, IEEE Transactions on Knowledge and Data Engineering 28 (4) (2015) 1007–1021.

[23] S. Bickel, T. Scheffer, Multi-view clustering., in: ICDM, Vol. 4, 2004, pp. 19–26.

[24] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on Machine Learning (ICML-11), 2011, pp. 393–400.

[25] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Advances in neural information processing systems, 2011, pp. 1413–1421.

[26] V. R. De Sa, P. W. Gallagher, J. M. Lewis, V. L. Malave, Multi-view kernel construction, Machine learning 79 (1-2) (2010) 47–71.

[27] S. Yu, L. Tranchevent, X. Liu, W. Glanzel, J. A. Suykens, B. De Moor, Y. Moreau, Optimized data fusion for kernel k-means clustering, IEEE Transactions on Pattern Analysis and Machine Intelligence 34 (5) (2012) 1031–1039.

[28] Y. Lu, L. Wang, J. Lu, J. Yang, C. Shen, Multiple kernel clustering based on centered kernel alignment, Pattern Recognition 47 (11) (2014) 3656–3664.

[29] G. Tzortzis, A. Likas, Kernel-based weighted multi-view clustering, in: 2012 IEEE 12th International Conference on Data Mining, IEEE, 2012, pp. 675–684.

[30] W. Tang, Z. Lu, I. S. Dhillon, Clustering with multiple graphs, in: 2009 Ninth IEEE International Conference on Data Mining, IEEE, 2009, pp. 1016–1021.

[31] S. F. Hussain, M. Mushtaq, Z. Halim, Multi-view document clustering via ensemble method, Journal of Intelligent Information Systems 43 (1) (2014) 81–99.

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

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

[34] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, IEEE transactions on pattern analysis and machine intelligence 35 (1) (2012) 171–184.

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

[36] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: 2009 IEEE Conference on Computer Vision and Pattern Recognition, IEEE, 2009, pp. 2790–2797.

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

[38] G. Liu, Z. Zhang, Q. Liu, H. Xiong, Robust subspace clustering with compressed data, IEEE Transactions on Image Processing.

[39] C.-G. Li, C. You, R. Vidal, On geometric analysis of affine sparse subspace clustering, IEEE Journal of Selected Topics in Signal Processing 12 (6) (2018) 1520–1533.

[40] H. Gao, F. Nie, X. Li, H. Huang, Multi-view subspace clustering, in: Proceedings of the IEEE international conference on computer vision, 2015, pp. 4238–4246.

[41] C. Zhang, Q. Hu, H. Fu, P. Zhu, X. Cao, Latent multi-view subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 4279–4287.

[42] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, L. Wang, Learning joint affinity graph for multi-view subspace clustering, IEEE Transactions on Multimedia.

[43] Y. Wang, W. Zhang, L. Wu, X. Lin, M. Fang, S. Pan, Iterative views agreement: an iterative low-rank based structured optimization method to multi-view spectral clustering, in: Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, AAAI Press, 2016, pp. 2153–2159.

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

[45] A. Y. Ng, M. I. Jordan, Y. Weiss, On spectral clustering: Analysis and an algorithm, in: Advances in neural information processing systems, 2002, pp. 849–856.

[46] Z. Kang, C. Peng, Q. Cheng, Z. Xu, Unified spectral clustering with optimal graph, in: Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18). AAAI Press, 2018.

[47] Z. Kang, H. Pan, S. C. H. Hoi, Z. Xu, Robust graph learning from noisy data, IEEE Transactions on Cybernetics (2019) 1–11`doi:10.1109/TCYB.2018.2887094`.

[48] Z. Zhang, J. Ren, S. Li, R. Hong, Z. Zha, M. Wang, Robust subspace discovery by block-diagonal adaptive locality-constrained representation, in: Proceedings of the 27th ACM International Conference on Multimedia, ACM, 2019, pp. 1569–1577.

[49] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 586–594.

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

[51] J. Wu, H. Liu, H. Xiong, J. Cao, J. Chen, K-means-based consensus clustering: A unified view, IEEE transactions on knowledge and data engineering 27 (1) (2014) 155–169.

[52] Z. Tao, H. Liu, S. Li, Z. Ding, Y. Fu, From ensemble clustering to multi-view clustering, in: IJCAI, 2017.

[53] X. Guo, Robust subspace segmentation by simultaneously learning data representations and their affinity matrix, in: Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015.

[54] H. Tao, C. Hou, X. Liu, T. Liu, D. Yi, J. Zhu, Reliable multi-view clustering, in: Thirty-Second AAAI Conference on Artificial Intelligence, 2018.

[55] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in: Twenty-Third International Joint conference on artificial intelligence, 2013.

[56] L. Zong, X. Zhang, L. Zhao, H. Yu, Q. Zhao, Multi-view clustering via multi-manifold regularized non-negative matrix factorization, Neural Networks 88 (2017) 74–89.

[57] Z. Kang, Z. Guo, S. Huang, S. Wang, W. Chen, Y. Su, Z. Xu, Multiple partitions aligned clustering, in: IJCAI, 2019, pp. 2701–2707.

[58] X. Gao, W. Hu, Z. Guo, Exploring graph learning for semi-supervised classification beyond euclidean data, arXiv preprint arXiv:1904.10146.

[59] S. A. Nene, S. K. Nayar, H. Murase, et al., Columbia object image library (coil-20).

[60] L. v. d. Maaten, G. Hinton, Visualizing data using t-sne, Journal of machine learning research 9 (Nov) (2008) 2579–2605.