Robust Graph-based Multi-view Clustering

Weixuan Liang,[1] Xinwang Liu,[1*] Sihang Zhou,[2] Jiyuan Liu,[1] Siwei Wang,[1] En Zhu,[1]

[1] College of Computer, National University of Defense Technology, Changsha, Hunan, China.
[2] College of Intelligence Science and Technology, National University of Defense Technology, Changsha, Hunan, China.
{weixuanliang,xinwangliu}@nudt.edu.cn, sihangjoe@gmail.com, {liujiyuan13,wangsiwei13,enzhu}@nudt.edu.cn

Abstract

Graph-based multi-view clustering (G-MVC) constructs a graphical representation of each view and then fuses them to a unified graph for clustering. Though demonstrating promising clustering performance in various applications, we observe that their formulations are usually non-convex, leading to a local optimum. In this paper, we propose a novel MVC algorithm termed robust graph-based multi-view clustering (RG-MVC) to address this issue. In particular, we define a min-max formulation for robust learning and then rewrite it as a convex and differentiable objective function whose convexity and differentiability are carefully proved. After that, we solve the resultant optimization problem using an efficient reduced gradient descent algorithm and prove the global optimality of the corresponding solution. As a consequence, although our algorithm is free of hyper-parameter, it has shown good robustness against noisy views. Extensive experiments on benchmark datasets verify the superiority of the proposed method against the compared state-of-the-art algorithms. Our codes and appendix are available at https://github.com/wx-liang/RG-MVC.

Introduction

Multi-view clustering (Yang and Wang 2018), which aims to fuse multiple views and learn a consensus representation for grouping unlabeled datasets into clusters, is an important research topic in the field of clustering. According to the information fusion strategy that is adopted, the existing algorithms can be roughly divided into five categories, i.e. co-training-based algorithms (Kumar, Rai, and Daume 2011), kernel-based algorithms (Zhao, Kwok, and Zhang 2009; Wang et al. 2021), subspace clustering-based algorithms (Gao et al. 2015; Zhou et al. 2020b,a), multi-task multi-view clustering (Zhang et al. 2016) and multi-view graph learning clustering (Nie, Li, and Li 2016). Among these methods, the graph learning-based methods have achieved considerable attention of researchers due to their superior capability of capturing the intrinsic cluster structure within data. The method that we studied also belongs to this category.

In recent years, numerous graph-based multi-view clustering algorithms are proposed to achieve high-quality partition with different fusion mechanism. (Nie, Li, and Li 2016) first constructs the base Laplacian matrices from different graphs, then combines them into an optimal Laplacian matrix for clustering, in which the combination coefficients are updated by the consensus clustering indicator matrix and basic Laplacian matrices. (Nie, Li, and Li 2017) learned a unified graph by minimizing the reconstruction error between the optimal graph and the weighted basic graph combination. (Zhan et al. 2019) merges the information from different graphs by first learning the cluster indicating matrices from individual views and then find their best linear combination to the similar matrix of a low rank Laplacian matrix. Through a graph diffusion method, (Tang et al. 2020) obtains a consensus graph that could effectively capture the complementary information from different views. By concatenating different graphs to a tensor, (Wu, Lin, and Zha 2019) dramatically improves the clustering performance through the tensor learning. Furthermore, (Wen et al. 2021) adjusts tensor framework to handle the datasets with incomplete views. (Huang et al. 2021) assumes that each graph can be divided into two parts, i.e., consistent part and divergent part. They fuse all consistent parts into a unified graph, while adopting the divergent parts to adjust the combination coefficients. In a recent work (Pan et al. 2021), the authors provide a novel algorithm by introducing contrastive learning for multi-view clustering and achieve good performance.

Although large performance enhancement has been achieved, we observe that the existing G-MVC algorithms usually follow a non-convex formulation. More preciously, the objective functions with multiple variables are not jointly convex. This property causes that most of the existing algorithms in this field are solved with an iterative optimization fashion, making them hard to guarantee the optimality of the corresponding solution. As a consequence, the performance of these algorithms are not fully exploited.

To solve the problem, this paper proposes a novel algorithm termed robust graph-based multi-view clustering (RG-MVC). Specifically, we first design a novel min-max optimization formulation to adversarially optimize the ideal consensus graph representation matrix and the graph combination coefficients in a unified framework. After that, we reformulate the min-max formulation into a differentiable and convex formulation and adopt the reduced gradient descent algorithm to solve the resulting optimization problem. Thanks to the adversarial learning mechanism and the con-

*Corresponding author

vexity of the revised formulation, our algorithm is guaranteed to converge to the global optimal solution and has shown good robustness against the graph-level noise.

The main contributions of the paper can be summarized as follows:

- We propose a novel convex graph-based multi-view clustering formulation. An efficient gradient descent-based algorithm is proposed to solve to resulting optimization problem.
- We give the strict proofs of the relevant properties, i.e., convexity and differentiability. Thus, the solution of our algorithm is global optimal.
- We conduct extensive experiments on seven benchmark datasets to verify the effectiveness and robustness of our proposed robust graph-based multi-view clustering algorithm.

Related Work

Graph-based Clustering

Graph-based clustering (GC) (Gan, Ma, and Wu 2007) is an important tool in the fields of clustering algorithms. After initializing a graph $\mathbf{S} \in \mathbb{R}^{n\times n}$, GC aims to partition this graph into k sub-graphs, where n is the sample number and k is the cluster number. The work in (Nie et al. 2016), which is termed Constrained Laplacian Rank (CLR), learns a rank constrained graph from initial graph $\mathbf{S}$. Specifically, CLR learns $\mathbf{G} \in \mathbb{R}^{n\times n}$ by:

$$\min_{\mathbf{G}} \|\mathbf{G}-\mathbf{S}\|_t \, s.t. \, \mathbf{G} \in \mathcal{C}, \tag{1}$$

where $\mathbf{S}$ denotes an initial graph, $\|\cdot\|_t$ denotes some norm, and $\mathcal{C} = \{\mathbf{G}|\mathbf{G}^\top\mathbf{1} = \mathbf{1}, G_{ij} \geq 0, \forall i,j \in [n], rank(\mathbf{L}_s) = n-k\}$ denotes the constraints of $\mathbf{G}$, where $\mathbf{L}_s = \mathbf{D}_s - (\mathbf{G}+\mathbf{G}^\top)/2$, $\mathbf{D}_s \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose i-th element is $\sum_{j=1}^{n}(G_{ij}+G_{ji})/2$. The output $\mathbf{G}$ in Eq. (1) has k connected components. Thus, clustering indicators can be obtained without post-processing on the graph. The constraint in Eq. (1) becomes mainstream in graph-based clustering research. However, the rank constraint also results in the feasible solution set being not convex, and the resulting objective function is non-convex. To make final optimization framework convex, we will relax the rank constraint, but retain the stochastic matrix constraint.

Graph-based Multi-view Clustering

In the multi-view setting, there are m basic graphs $\{\mathbf{S}_p\}_{p=1}^m \subset \mathbb{R}^{n\times n}$. Graph-based multi-view clustering (G-MVC) aims to fuse these basic graphs into a consensus graph. The work in (Nie, Li, and Li 2017) learns a rank constrained graph as follows:

$$\min_{\mathbf{G}} \sum_{p=1}^{m} \gamma_p \|\mathbf{G}-\mathbf{S}_p\|_{\mathrm{F}}^2 \, s.t. \, \mathbf{G} \in \mathcal{C} \tag{2}$$

where $\mathcal{C} = \{\mathbf{G}|\mathbf{G}^\top\mathbf{1} = \mathbf{1}, G_{ij} \geq 0, \forall i,j \in [n], rank(\mathbf{L}_s) = n-k\}$ denotes the constraints of $\mathbf{G}$, and γ_p is the weight of the p-th view which is given by: $\gamma_p = 1/2\left(\|\mathbf{G}-\mathbf{S}_p\|_{\mathrm{F}}^2\right)$.

Due to the rank constraint, the objective function in Eq. (2) is also non-convex. Thus, the relevant problem is solved by an alternative optimization, and the optimal solution may not be discovered. There are also some G-MVC algorithms (Tao et al. 2017; Liang, Huang, and Wang 2019) which are free from the rank constraint. However, almost all of these methods fall into alternative optimization, resulting in the exact optimal solution that cannot guarantee to be obtained.

Proposed Method

The Proposed Formulation

Given m basic graphs $\{\mathbf{S}_p\}_{p=1}^m \subset \mathbb{R}^{n\times n}$, we assume that the optimal graph $\mathbf{S}_\gamma = \sum_{p=1}^{m} \gamma_p^2 \mathbf{S}_p$ is the combination of $\{\mathbf{S}_p\}_{p=1}^m \subset \mathbb{R}^{n\times n}$, where γ_p is the combining coefficient of the p-th view. With fixed $\boldsymbol{\gamma}$, we aim to learn a graph $\mathbf{G}$ by maximizing the alignment between the optimal graph and the combined base graphs as follows:

$$\max_{\mathbf{G}} \mathrm{Tr}(\mathbf{G}^\top\mathbf{S}_\gamma) \, s.t. \mathbf{G} \in \mathcal{G} \tag{3}$$

In this formulation $\mathcal{G} = \{\mathbf{G}|\mathbf{G}^\top\mathbf{1} = \mathbf{1}, G_{ij} \geq 0, G_{ii} = 0, \forall i,j \in [n]\}$, it follows a stochastic matrix constraint. Moreover, the diagonal values of the solution are required to be 0. To avoid the trivial solution, we further modify Eq.(3) by adding a regularization term as follow:

$$\begin{aligned} &\max_{\mathbf{G}\in\mathcal{G}} f(\mathbf{G},\boldsymbol{\gamma}) \\ &f(\mathbf{G},\boldsymbol{\gamma}) = \mathrm{Tr}(\mathbf{G}^\top\mathbf{S}_\gamma) - \|\mathbf{G}\|_{\mathrm{F}}^2 \end{aligned} \tag{4}$$

Given an ideal consensus graph $\mathbf{G}$, we further find the optimal coefficient combination by optimizing Eq. (5).

$$\min_{\boldsymbol{\gamma}\in\triangle} \sum_{p=1}^{m} \gamma_p^2 A_p, \tag{5}$$

where $A_p = \mathrm{Tr}(\mathbf{G}^\top\mathbf{S}_p)$ and $\triangle = \{\boldsymbol{\gamma}|\boldsymbol{\gamma}^\top\mathbf{1} = 1, \gamma_p \geq 0, \forall p \in [m]\}$. In our setting, we minimize Eq. (5) w.r.t. the combination weights. By this way, 1) we can keep more diverse information by assigning a large coefficient to a graph with relatively small alignment value; 2) we can forbid the consensus graph to get too close to the noisy graphs, thus making the algorithm more robust to graph-level noise information.

Taken the above parts into consideration, we have the following min-max optimization formulation:

$$\begin{aligned} &\min_{\boldsymbol{\gamma}\in\triangle} \max_{\mathbf{G}\in\mathcal{G}} f(\mathbf{G},\boldsymbol{\gamma}) \\ &\text{s.t. } f(\mathbf{G},\boldsymbol{\gamma}) = \mathrm{Tr}(\mathbf{G}^\top\mathbf{S}_\gamma) - \|\mathbf{G}\|_{\mathrm{F}}^2 \end{aligned} \tag{6}$$

Although carefully designed, the above formulation is hard to be optimized by the commonly adopted iterative optimization algorithm. To this end, we further transfer the min-max formulation into the following equivalent form:

$$\begin{aligned} &\min_{\boldsymbol{\gamma}\in\triangle} F(\boldsymbol{\gamma}) \\ &\text{s.t. } F(\boldsymbol{\gamma}) = \max_{\mathbf{G}\in\mathcal{G}} \mathrm{Tr}(\mathbf{G}^\top\mathbf{S}_\gamma) - \|\mathbf{G}\|_{\mathrm{F}}^2 \end{aligned} \tag{7}$$

The new formulation has two desirable properties, i.e.,**differentiability and convexity**. As a result, we solve the optimization problem in Eq. (7) with a gradient descent algorithm and prove the global optimality of the obtained solution.

Theoretical Analysis

In this subsection, we provide the proofs of the differentiability and convexity of $F(\boldsymbol{\gamma})$.

Theorem 1 (Differentiability)**.** *$F(\boldsymbol{\gamma})$ in Eq. (7) is differentiable. Specifically,*

$$\frac{\partial F(\boldsymbol{\gamma})}{\partial \gamma_p} = 2\gamma_p \mathrm{Tr}(\hat{\mathbf{G}}^\top \mathbf{S}_p),$$

where $\hat{\mathbf{G}} = \mathrm{argmax}_{\mathbf{G}\in\mathcal{G}} \mathrm{Tr}(\mathbf{G}^\top \mathbf{S}_{\boldsymbol{\gamma}}) - \|\mathbf{G}\|_F^2$

We aim to prove Theorem 1 by Danskin's Theorem (Danskin 1966). Before proving it, we need to prove the following three lemmas about the properties of the unified graph $\mathbf{G}$, i.e., the uniqueness, the compactness of the feasible solution set and the continuity.

Lemma 1 (Uniqueness)**.** *With fixed $\boldsymbol{\gamma}$, the solution of*

$$\max_{\mathbf{G}\in\mathcal{G}} \mathrm{Tr}(\mathbf{G}^\top \mathbf{S}_{\boldsymbol{\gamma}}) - \|\mathbf{G}\|_F^2 \tag{8}$$

is unique.

Proof. For the ease of understanding, we first provide the solution of Eq. (8) and prove its uniqueness, the detailed deduction can be found in the next algorithm optimization subsection. Denote $\mathbf{G} = [\mathbf{g}_1, ..., \mathbf{g}_n]$, where $\mathbf{g}_i$ is the i-th column of $\mathbf{G}$, $\mathbf{g}_i$ has a closed-form solution as follows:

$$\hat{g}_{ij} = \begin{cases} \max(\frac{s_{ij}}{2} + \eta, 0) & if\ j \neq i, \\ 0 & if\ j = i, \end{cases} \tag{9}$$

where η is any real number that makes $\hat{\mathbf{g}}_i$ satisfy the constraint $\mathcal{G}$. Lemma 1 would hold if we can prove that the η in Eq. (9) is unique. We use reduction to absurdity to acquire the conclusion.

Assume that there are two real numbers $\eta_1 \neq \eta_2$ which can make $\mathbf{g}_i^\top \mathbf{1} = 1$ hold. Without loss of generality, we assume that $\eta_1 > \eta_2$. Denote that $s'_1, ..., s'_n$ is the permutation of $s_{1j}, ..., s_{nj}$ in ascending order. There exists two integers p, q such that $\frac{s'_p}{2} + \eta_1 > 0$, $\frac{s'_{p+1}}{2} + \eta_1 < 0$, $\frac{s'_q}{2} + \eta_1 > 0$, and $\frac{s'_{q+1}}{2} + \eta_1 < 0$ hold, simultaneously. Because $\eta_1 > \eta_2$, we have $s'_p > s'_q$, i.e., $p \geq q$.

When $p = q$,

$$\sum_{i=1}^{p}(\frac{s'_i}{2} + \eta_1) = \sum_{i=1}^{q}(\frac{s'_i}{2} + \eta_2) = 1.$$

We have $\eta_1 = \eta_2$, and it is in contradiction with the assumption.

When $p > q$,

$$\begin{aligned} 1 = \sum_{i=1}^{p}(\frac{s'_i}{2} + \eta_1) &> \sum_{i=1}^{q}\frac{s'_i}{2} + p\eta_1 \\ &> \sum_{i=1}^{q}\frac{s'_i}{2} + q\eta_2 = \sum_{i=1}^{q}(\frac{s'_i}{2} + \eta_2) = 1. \end{aligned}$$

This is also in contradiction. As a consequence, there does not exist two different ηs which meet the constraint in Eq. (9). This proves Lemma 1. □

Lemma 2 (Compactness)**.** *$\mathcal{G}$ is compact.*

Proof. Denote function $p : \mathbb{R}^{n\times n} \to \mathbb{R}^n$ as

$$p(\mathbf{C}) = \mathbf{C}^\top \mathbf{1} - \mathbf{1}.$$

Because p is continuous and $\forall \mathbf{G} \in \mathcal{G}$, $p(\mathbf{G}) = \mathbf{0}$, we can obtain that $\mathcal{G}$ is closed. Moreover, $\mathcal{G}$ is bounded. This proves the result. □

Lemma 3 (Continuity)**.** *Define the function (w.r.t $\boldsymbol{\gamma}$) $G : \mathbb{R}^n \to \mathbb{R}^{n\times n}$ as*

$$G(\boldsymbol{\gamma}) = \operatorname*{argmax}_{\mathbf{G}\in\mathcal{G}} f(\mathbf{G}, \boldsymbol{\gamma}),$$

where $f(\mathbf{G}, \boldsymbol{\gamma}) = \mathrm{Tr}(\mathbf{G}^\top \mathbf{S}_{\boldsymbol{\gamma}}) - \|\mathbf{G}\|_F^2$. Then, $G(\boldsymbol{\gamma})$ is continuous.

The proof of Lemma 3 is omitted in the main text due to space limited. Please refer to appendix for the proof. We give the proof of Theorem 1 as follows.

Proof. According to Lemma 1, Lemma 2 and Lemma 3, it can be checked that all the conditions of Danskin's Theorem (Danskin 1966) hold. Thus, $F(\boldsymbol{\gamma})$ is differentiable. □

Theorem 2 (Convexity)**.** *$F(\boldsymbol{\gamma})$ is a convex function.*

We give the proof of Theorem 2 in appendix due to the space limited.

Remark. *By Theorem 1, we know that $F(\boldsymbol{\gamma})$ is differentiable. Thus, we can optimize $F(\boldsymbol{\gamma})$ by gradient descent algorithm, and the objective function will convergence to global minimum by Theorem 2.*

The Optimization Algorithm

We first introduce how to optimize $\mathbf{G}$ with fixed $\boldsymbol{\gamma}$ in function $F(\boldsymbol{\gamma})$. Denote $\mathbf{G} = [\mathbf{g}_1, ..., \mathbf{g}_n]$, where $\mathbf{g}_i$ is the i-th column of $\mathbf{G}$. It is easy to verify that $\{\mathbf{g}_i\}_{i=1}^n$ are pairwise independent. Thus, $\mathbf{G}$ can be optimized column-by-column as follows:

$$\begin{aligned} &\min_{\mathbf{g}_i} \mathbf{g}_i^\top \mathbf{s}_i - \mathbf{g}_i^\top \mathbf{g}_i \\ &s.t.\, \mathbf{g}_i^\top \mathbf{1} = 1,\, g_{ii} = 0,\, g_{ij} \geq 0,\, (\forall j \in [n]). \end{aligned} \tag{10}$$

where $\mathbf{s}_i$ is the i-th column of $\mathbf{S}_{\boldsymbol{\gamma}}$. By following (Zhan et al. 2019), the optimal solution $\hat{\mathbf{g}}_i$ of Eq. (10) is

$$\hat{g}_{ij} = \begin{cases} \max(\frac{s_{ij}}{2} + \eta, 0) & if\ j \neq i, \\ 0 & if\ j = i, \end{cases} \tag{11}$$

where η is any real number that makes $\hat{\mathbf{g}}_i$ satisfy the constraint $\mathcal{G}$ and it is easy to obtain by a search algorithm.

Then, we will describe how to optimize $F(\boldsymbol{\gamma})$. By following (Rakotomamonjy et al. 2008), we adopt a reduced gradient descent algorithm to optimize Eq. (7).

By Theorem 1, we can calculate the gradient of $F(\boldsymbol{\gamma})$. The main difficulty is how to update $\boldsymbol{\gamma}$ with this gradient while keeping the equality and non-negativity constraints. Denote

$\nabla F(\boldsymbol{\gamma})$ as the reduced gradient of $F(\boldsymbol{\gamma})$ and u is the index of $\boldsymbol{\gamma}$'s largest component. The p-th ($p \in [m]$) element of $\nabla F(\boldsymbol{\gamma})$ is

$$[\nabla F(\boldsymbol{\gamma})]_p = \frac{\partial F(\boldsymbol{\gamma})}{\partial \gamma_p} - \frac{\partial F(\boldsymbol{\gamma})}{\partial \gamma_u} \ \forall p \neq u$$

and

$$[\nabla F(\boldsymbol{\gamma})]_u = \sum_{p=1, p\neq u}^{m} \left(\frac{\partial F(\boldsymbol{\gamma})}{\partial \gamma_u} - \frac{\partial F(\boldsymbol{\gamma})}{\partial \gamma_p} \right)$$

Thus, updating by this gradient, the constraint $\boldsymbol{\gamma}^\top \mathbf{1} = 1$ can be satisfied. To meet the positive constraints, we set the descent direction as

$$d_p = \begin{cases} 0 & if\ \gamma_p = 0 \,\&\, [\nabla F(\boldsymbol{\gamma})]_p > 0 \\ -[\nabla F(\boldsymbol{\gamma})]_p, & if\ \gamma_p > 0 \,\&\, p \neq u \\ -[\nabla F(\boldsymbol{\gamma})]_u, & if\ p = u \end{cases} \tag{12}$$

Denote $\mathbf{d} = [d_1, ..., d_m]$ and α is the learning step, we can update $\boldsymbol{\gamma}$ as $\boldsymbol{\gamma} = \boldsymbol{\gamma} + \alpha \mathbf{d}$. The optimal α can be decided by Armijo's rule.

Obtain the Final Clustering Results

Through the optimization of Eq. (7), we obtain a graph that fuses the information of all views. In this subsection, we will introduce how to obtain the final clustering results. The obtained graph $\mathbf{G}$ may not be a symmetric matrix. Thus we perform the standard spectral clustering on $\frac{\mathbf{G}+\mathbf{G}^\top}{2}$ for the final clustering results.

The complete procedure of the proposed RG-MVC is summarized in Algorithm 1.

Algorithm 1: The proposed RG-MVC

Input: Pre-defined graph matrices $\{\mathbf{S}_p\}_{p=1}^m$, cluster number k

1: initialize $\boldsymbol{\gamma}^{(1)} = \mathbf{1}/m$, $t = 1$;
2: **while** not converge **do**
3: compute $\hat{\mathbf{G}}$ by solving Eq. (10).
4: compute $\frac{\partial F(\boldsymbol{\gamma})}{\partial \gamma_p}(p \in [m])$ and the descent direction $\mathbf{d}^{(t)}$ according to Eq (12).
5: $\boldsymbol{\gamma}^{(t+1)} \leftarrow \boldsymbol{\gamma}^{(t)} + \alpha \mathbf{d}^{(t)}$.
6: $t \leftarrow t + 1$.
7: **end while**
8: Obtain clustering results by standard spectral clustering on $\frac{\mathbf{G}+\mathbf{G}^\top}{2}$.

Output: Clustering results.

Computational Complexity and Convergence

Computational complexity. As shown in Algorithm 1, during each iteration, it has three steps: computing $\hat{\mathbf{G}}$, computing the corresponding reduced gradient and searching the optimal step size. Obtaining $\hat{\mathbf{G}}$ needs to solve n optimization problems on the n-dimensional simplex space. In total, these three steps cost $\mathcal{O}(n^2 + mn^2 + ml)$ time, where l is the max number of steps to obtain the optimal α. At last, the time consumption of the standard spectral clustering is $\mathcal{O}(n^3)$.

Convergence. By following Theorem 2, we can get that $F(\boldsymbol{\gamma})$ is convex. Thus, the objective function will converge to the global minimum by the reduced gradient descent algorithm.

Table 1: Benchmark datasets

Datasets	Samples	Views	Clusters
Flo17	1360	7	17
Flo102	8189	4	102
DIGIT	2000	3	10
Mfeat	2000	12	10
Cal102	6773	3	20
YALE	165	5	15
PFold	694	12	27

Experiments

In this section, we conduct experiments to verify the effectiveness of the proposed RG-MVC. Specifically, the clustering performance, algorithm convergence and the robustness against the graph-level noise is tested to conduct the validation.

Experimental Settings

Seven benchmark datasets are adopted to demonstrate the effectiveness of the proposed method, including *Flo17*[1], *Flo102*[2], *DIGIT*[3], *Mfeat*[4], *Cal102*[5], *PFold*[6] and *YALE*[7]. All graph matrices of these datasets are pre-computed and widely used in graph-based MVC and multiple kernel clustering. The detailed information is listed in Table 1. As seen, the numbers of samples, views and clusters vary over a large range. As a result, we can evaluate the performance of different algorithms comprehensively. For all experiments, we set the number of clusters to the true class number of the corresponding dataset. Three widely used metrics, i.e., accuracy (ACC), normalized mutual information (NMI) and purity, are adopted to verify the clustering performance. To get rid of the adverse effect of the randomness of k-means clustering evaluation, we repeat this process for 50 times and record their average values as final clustering results. All the experiments are conducted on a desktop computer with Intel(R) Core(TM)-i7-7820X CPU and 64G RAM.

Comparison with state-of-the-art algorithms

To evaluate the clustering performance of the proposed method, RG-MVC is further compared with following state-of-the-art multi-view clustering algorithms.

[1] www.robots.ox.ac.uk/~vgg/data/flowers/17/
[2] www.robots.ox.ac.uk/~vgg/data/flowers/102/
[3] http://ss.sysu.edu.cn/py/
[4] https://archive.ics.uci.edu/ml/datasets/Multiple+Features
[5] www.vision.caltech.edu/Image_Datasets/Caltech101/
[6] mkl.ucsd.edu/dataset/protein-fold-prediction
[7] http://vision.ucsd.edu/content/yale-face-database

Table 2: Empirical evaluation and comparison of RG-MVC with nine baseline methods on 7 benchmark datasets in terms of clustering accuracy (ACC), normulaized mutual information (NMI) and Purity.

datasets	Avg-SC	SB-SC	MKKM	MKKM-MR	RMKKM	MLAN	AMGL	MCGC	SMKKM	RG-MVC
ACC(%)										
Flo17	56.25	42.57	42.35	58.24	49.19	51.03	56.32	58.31	61.03	**67.86**
Flo102	34.63	36.02	21.87	40.02	29.39	23.53	33.35	39.05	41.96	**44.02**
DIGIT	90.30	72.45	47.00	90.75	40.75	96.20	92.85	92.75	90.60	**96.45**
Mfeat	93.55	85.85	54.80	94.60	63.45	**97.35**	84.35	95.60	94.15	96.95
Cal102	34.51	33.01	33.01	36.86	30.65	25.88	37.65	28.76	34.77	**38.03**
YALE	60.61	56.97	55.15	60.61	57.58	58.18	60.00	63.03	60.61	**63.63**
PFold	31.84	36.02	26.80	35.45	33.86	29.39	36.89	36.17	34.87	**39.76**
Average	57.38	51.84	40.14	59.50	43.55	54.51	57.34	59.10	59.71	**63.81**
NMI(%)										
Flo17	55.73	45.39	44.06	56.80	50.93	56.39	56.98	59.60	58.93	**67.52**
Flo102	52.03	48.49	42.23	56.97	48.84	34.78	51.63	50.72	58.26	**60.69**
DIGIT	83.56	64.73	49.02	83.75	47.65	91.88	86.65	85.65	83.57	**92.20**
Mfeat	87.73	75.62	52.05	89.12	66.02	**94.23**	81.58	90.71	88.70	93.42
Cal102	60.10	58.99	58.94	60.99	54.29	42.43	61.79	62.50	60.22	**62.65**
YALE	60.43	58.42	55.84	60.09	59.95	58.61	61.20	**63.46**	60.28	62.23
PFold	41.62	41.61	38.51	44.26	41.35	28.33	44.19	36.72	44.45	**48.88**
Average	63.03	56.18	48.66	64.57	52.72	58.09	63.43	64.19	64.92	**69.66**
Purity(%)										
Flo17	57.79	45.44	43.53	59.41	50.96	55.44	58.16	61.91	61.62	**70.07**
Flo102	39.88	40.78	27.52	46.17	33.56	30.61	39.71	46.84	48.66	**50.87**
DIGIT	90.30	72.45	50.20	90.75	45.95	96.20	92.85	92.75	90.60	**96.45**
Mfeat	93.55	85.85	56.40	94.60	67.20	**97.35**	84.35	95.60	94.15	96.95
Cal102	36.67	35.42	35.62	39.35	32.35	28.17	39.28	40.11	37.19	**40.58**
YALE	60.61	58.79	56.36	60.61	58.18	58.79	60.61	63.64	60.61	**64.24**
PFold	38.62	40.78	35.73	41.64	38.62	33.00	42.07	39.48	43.23	**48.12**
Average	59.63	54.22	43.62	61.79	46.69	57.08	59.58	62.90	62.29	**66.75**

- **Average spectral clustering (Avg-SC)**: It takes the average graph as the input for the standard spectral clustering algorithm.
- **Single best spectral clustering (SB-SC)**: Standard spectral clustering algorithm is performed on each single graph and the best result is reported.
- **Multiple kernel k-means (MKKM)** (Huang, Chuang, and Chen 2012): The algorithm performs kernel k-means and combination coefficients optimization simultaneously within a unified framework.
- **Multiple kernel k-means with matrix-induced regularization (MKKM-MR)** (Liu et al. 2016): This algorithm introduces a matrix-induced regularization to reduce the redundancy and enhance the diversity of the combined kernels.
- **Robust multiple kernel k-means (RMKKM)** (Du et al. 2015): RMKKM learns a robust low-rank kernel for clustering by filtering the noise structures in multiple kernels.
- **Multi-view learning with adaptive neighbors (MLAN)** (Nie, Cai, and Li 2017): MLAN constructs a consensus graph by an adaptive neighbor approach, while performing clustering by a unified framework.
- **Auto-weighted multiple graph learning (AMGL)** (Nie, Li, and Li 2016): AMGL learns the combination coefficients of each graph automatically via the reformulation of standard spectral clustering.
- **Multiview consensus graph clustering (MCGC)** (Zhan et al. 2019): By learning the graph and the embedding matrices simultaneously, MCGC obtains a consensus graph with desirable clustering structure.
- **Simple multiple Kernel k-means (SMKKM)** (Liu, Zhu, and Liu 2020): SMKKM re-formulates MKKM as a min-max problem in the kernel coefficients and the consensus clustering indicator matrix.

To achieve the best performance of the compared algorithms, to those algorithms with hyper-parameters, we perform grid search on the parameters suggested by the authors and report their best results.

Experimental Results

Table 2 reports the total clustering results on seven benchmark datasets. The best value is marked in bold. From Table 2, we observe that:

- The proposed algorithm is superior to all comparison methods. Including all benchmark datasets, RG-MVC averagely exceeds the second best algorithms by 4.10%, 4.74% and 3.85% in terms of ACC, NMI and purity.
- As a strong baseline, SMKKM (Liu, Zhu, and Liu 2020) achieves high performance in comparison with most MVC algorithms as shown in Table 2. However, the proposed RG-MVC consistently outperforms SMKKM by **6.83%**, **2.06%**, **5.85%**, **2.80%**, **3.26%**, **3.03%** and **4.89%** in terms of ACC on the benchmark datasets.

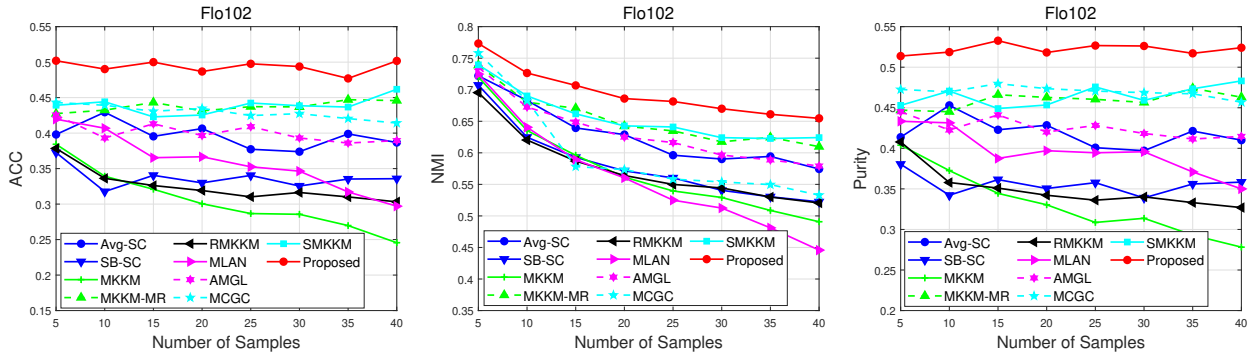

Figure 1: ACC, NMI and purity comparison with variation of sample numbers on Flo102. These datasets are constructed by the first 5,10,...,40 samples of each class from Flo102.

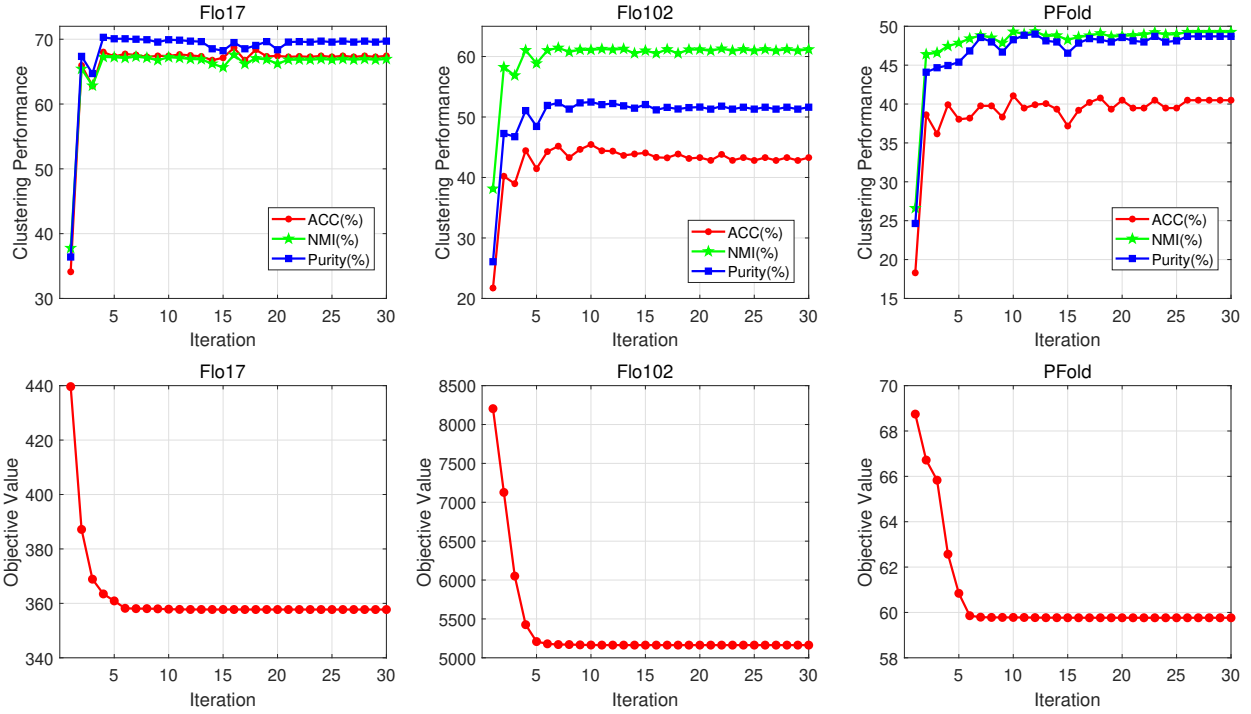

Figure 2: Illustration of performance variation and algorithm convergence. The upper row illustrates the performance variation on Flo17, Flo102 and PFold as the iteration increases. The bottom row illustrates the corresponding objective value.

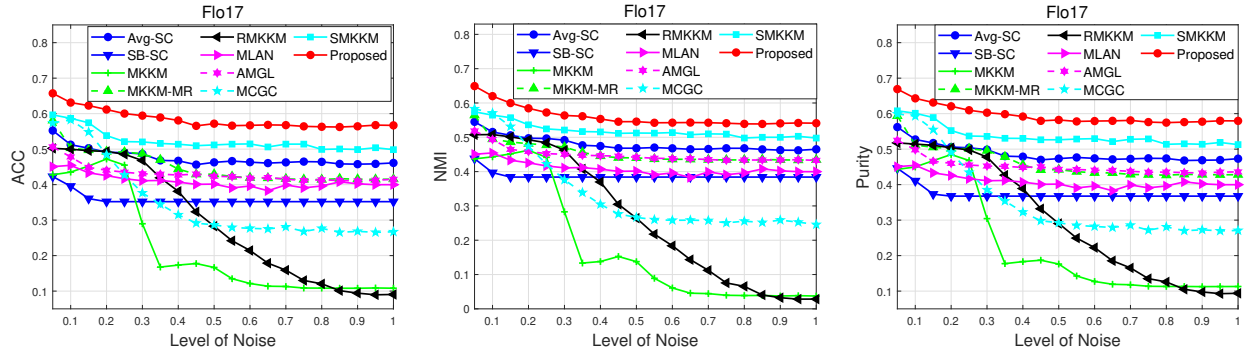

Figure 3: ACC, NMI and purity comparison on Flo17 with different noise level. The noise level ranges in $\{0.05,0.1,...,1\}$.

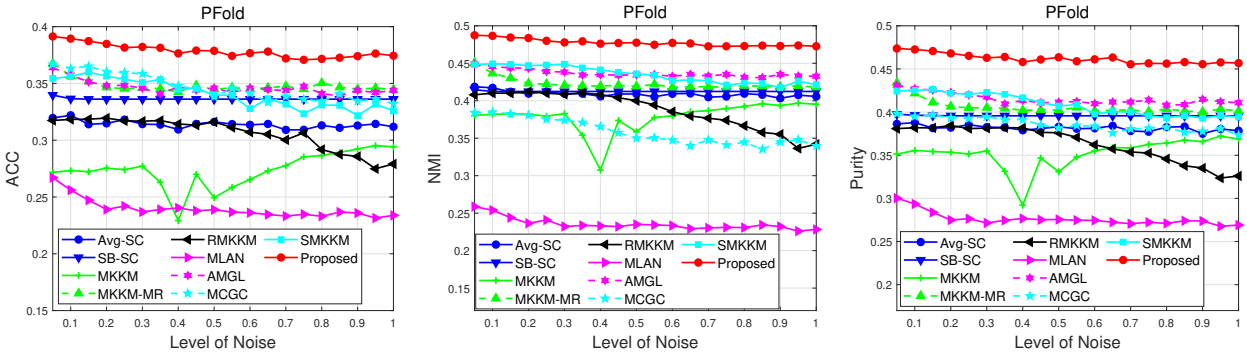

Figure 4: ACC, NMI and purity comparison on PFold with different noise level. The noise level ranges in $\{0.05,0.1,...,1\}$.

- As a strong baseline of graph-based MVC, MCGC also achieves high clustering performance. However, MCGC needs to select a hyper-parameter and this will limit its real-world application due to the lack of ground truth in clustering tasks. By contrast, the proposed RG-MVC is parameter-free and outperforms MCGC by **9.55%**, **4.97%**, **3.70%**, **1.35%**, **9.27%**, **0.60%** and **3.59%** in terms of ACC on the benchmark datasets.

Moreover, eight sub-datasets constructed by the first $5, 10, ..., 40$ samples of each class from Flo102, are also used in our experiments. To visualize the effectiveness of RG-MVC, we illustrate the clustering results in Figure 1. The red curve represents the proposed RG-MVC. As seen, the proposed RG-MVC consistently achieves the best performance at all sample numbers.

Algorithm Convergence

To evaluate the learning effectiveness, we record the variation objective value and corresponding clustering performance along with iterations. Due to space limited, we only show the results on three datasets, i.e., Flo17, Flo102 and PFold, in Figure 2. It can be observed that the objective function monotonically decreases and reaches convergence within 10 iterations. In addition, the clustering performance increases in the several forward iterations, then slightly fluctuates and keeps steady until the objective value converges. In general, Figure 2 clearly demonstrates the convergence of the optimization algorithm and the effectiveness of the learned consensus graph.

Robustness to Noise

We conduct experiments on Flo17 and PFold with noises to verify the robustness of the proposed algorithm. Specifically, we add different levels of Gaussian noise to partial views. In even-valued views, we add standard Gaussian noises (i.e., expectation $\mu = 0$ and standard deviation $\delta = 1$) with a multiplying factor α to 80 % items. The rest views keep unchanged. We document the variations of clustering results when α ranges in $\{0.05, 0.1, ..., 1\}$. To reduce the influences of randomness, we run all experiments 30 times and adopt the average values. Figure 3 and 4 illustrate the results on Flo17 and PFold, respectively, and the red curve denotes the proposed algorithm. It can be observed that the proposed method outperforms all the comparison methods. As illustrated in Figure 3, the performance of MKKM, RMKKM, and MCGC rapidly decreases with the increase of noises, while the others vary slightly. In terms of ACC, NMI, and purity, the proposed RG-MVC averagely exceeds SMKKM, which is the second-best on Flo17 by 6.14%, 4.11% and 5.82% with different noise level. In Figure 4, the curves of AMGL, MKKM and RMKKM fluctuate dramatically. At the same time, we can see that MKKM-MR, AMGL and SMKKM have similar performances, but they are all inferior to our algorithm. Moreover, the weight variation of RG-MVC on Flo17 without noise, $\alpha = 0.5$ and $\alpha = 1$. As observed in Figure 5, the weights of noisy even-numbered views reduce, while the weights of noise-free odd-numbered views increase. Thus, the robustness w.r.t. views of RG-MVC can be guaranteed.

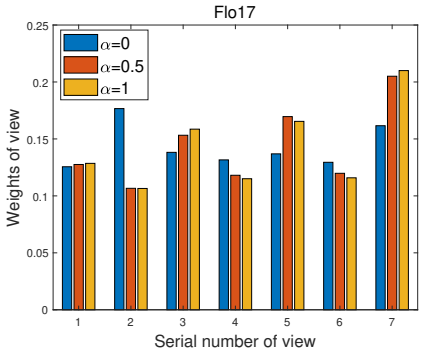

Figure 5: Weight variation of RG-MVC on Flo17 with different noise level.

Conclusion

In this paper, we propose an effective, robust and parameter-free method, which is termed graph-based multi-view clustering (RG-MVC). In particular, we define a min-max formulation for robust learning and then rewrite it as a convex and differentiable objective function whose convexity and differentiability are carefully proved. A reduced gradient descent algorithm is adopted to solve the relevant minimization question. Finally, we conduct extensive experiments on seven benchmark datasets to verify the effectiveness and robustness of the proposed RG-MVC. In the future, we plan to reduce the computational complexity of the proposed method for the application on large-scale datasets.

Acknowledgments

This work was supported by the National Key R&D Program of China (project no. 2020AAA0107100) and the National Natural Science Foundation of China (project no. 61922088, 61906020, 61872371 and 62006237).

References

Danskin, J. M. 1966. The Theory of Max-Min, with Applications. In *SIAM Journal on Applied Mathematics*, 641–664.

Du, L.; Zhou, P.; Shi, L.; Wang, H.; Fan, M.; Wang, W.; and Shen, Y.-D. 2015. Robust Multiple Kernel K-Means Using 2;1-Norm. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 3476–3482.

Gan, G.; Ma, C.; and Wu, J. 2007. Data clustering - theory, algorithms, and applications. I–XXII, 1–466. SIAM.

Gao, H.; Nie, F.; Li, X.; and Huang, H. 2015. Multi-view Subspace Clustering. In *IEEE International Conference on Computer Vision (ICCV)*, 4238–4246.

Huang, H.-C.; Chuang, Y.-Y.; and Chen, C.-S. 2012. Multiple Kernel Fuzzy Clustering. In *IEEE Transactions on Fuzzy Systems (TFS)*, 120–134.

Huang, S.; Tsang, I.; Xu, Z.; and Lv, J. C. 2021. Measuring Diversity in Graph Learning: A Unified Framework for Structured Multi-view Clustering.

Kumar, A.; Rai, P.; and Daume, H. 2011. Co-regularized Multi-view Spectral Clustering. In *Advances in Neural Information Processing Systems (NeurIPS)*, 1413–1421.

Liang, Y.; Huang, D.; and Wang, C.-D. 2019. Consistency Meets Inconsistency: A Unified Graph Learning Framework for Multi-view Clustering. In *2019 IEEE International Conference on Data Mining (ICDM)*, 1204–1209.

Liu, X.; Dou, Y.; Yin, J.; Wang, L.; and Zhu, E. 2016. Multiple Kernel k-Means Clustering with Matrix-Induced Regularization. In *AAAI Conference on Artificial Intelligence (AAAI)*, 1888–1894.

Liu, X.; Zhu, E.; and Liu, J. 2020. SimpleMKKM: Simple Multiple Kernel K-means. In *CoRR, abs/2005.04975*.

Nie, F.; Cai, G.; and Li, X. 2017. Multi-View Clustering and Semi-Supervised Classification with Adaptive Neighbours. In *AAAI Conference on Artificial Intelligence (AAAI)*, 2408–2414.

Nie, F.; Li, J.; and Li, X. 2016. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 1881–1887.

Nie, F.; Li, J.; and Li, X. 2017. Self-Weighted Multiview Clustering with Multiple Graphs. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2564–2570.

Nie, F.; Wang, X.; Jordan, M. I.; and Huang, H. 2016. The Constrained Laplacian Rank Algorithm for Graph-Based Clustering. In *International Joint Conference on Artificial Intelligence (AAAI)*, 1969–1976.

Pan, E.; et al. 2021. Multi-view Contrastive Graph Clustering. In *Thirty-Fifth Conference on Neural Information Processing Systems (NeurIPS)*.

Rakotomamonjy, A.; Bach, F.; Canu, S.; Grandvalet, Y.; et al. 2008. SimpleMKL. In *Journal of Machine Learning Research (JMLR)*, 2491–2521.

Tang, C.; Liu, X.; Zhu, X.; Zhu, E.; Luo, Z.; Wang, L.; and Gao, W. 2020. CGD: Multi-View Clustering via Cross-View Graph Diffusion. In *AAAI Conference on Artificial Intelligence (AAAI)*, 5924–5931.

Tao, Z.; Liu, H.; Li, S.; Ding, Z.; and Fu, Y. 2017. From Ensemble Clustering to Multi-View Clustering. In *International Joint Conference on Artificial Intelligence (IJCAI)*, 2843–2849.

Wang, S.; Liu, X.; Liu, L.; Zhou, S.; and Zhu, E. 2021. Late Fusion Multiple Kernel Clustering With Proxy Graph Refinement. In *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*.

Wen, J.; Zhang, Z.; Zhang, Z.; Zhu, L.; Fei, L.; Zhang, B.; and Xu, Y. 2021. Unified Tensor Framework for Incomplete Multi-view Clustering and Missing-view Inferring. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 10273–10281.

Wu, J.; Lin, Z.; and Zha, H. 2019. Essential Tensor Learning for Multi-View Spectral Clustering. In *IEEE Transactions on Image Processing (TIP)*, 5910–5922.

Yang, Y.; and Wang, H. 2018. Multi-view clustering: A survey. In *Big Data Mining and Analytics*, 83–107.

Zhan, K.; Nie, F.; Wang, J.; and Yang, Y. 2019. Multiview Consensus Graph Clustering. In *IEEE Transactions on Image Processing (TIP)*, 1261–1270.

Zhang, X.; Zhang, X.; Liu, H.; and Liu, X. 2016. Multi-Task Multi-View Clustering. In *IEEE Transactions on Knowledge and Data Engineering (TKDE)*, 3324–3338.

Zhao, B.; Kwok, J. T.; and Zhang, C. 2009. Multiple Kernel Clustering. In *SIAM International Conference on Data Mining (SDM)*, 638–649.

Zhou, S.; Liu, X.; Li, M.; Zhu, E.; Liu, L.; Zhang, C.; and Yin, J. 2020a. Multiple Kernel Clustering With Neighbor-Kernel Subspace Segmentation. In *IEEE Transactions on Neural Networks and Learning Systems (TNNLS)*, 1351–1362.

Zhou, S.; Zhu, E.; Liu, X.; Zheng, T.; Liu, Q.; Xia, J.; and Yin, J. 2020b. Subspace segmentation-based robust multiple kernel clustering. In *Information Fusion*, 145–154.