Consistency Meets Inconsistency: A Unified Graph Learning Framework for Multi-view Clustering

Youwei Liang[†], Dong Huang[†]*, Chang-Dong Wang[‡]
[†]*College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China*
[‡]*School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China*
liangyouwei1@gmail.com, huangdonghere@gmail.com, changdongwang@hotmail.com

***Abstract*—Graph Learning has emerged as a promising technique for multi-view clustering, and has recently attracted lots of attention due to its capability of adaptively learning a unified and probably better graph from multiple views. However, the existing multi-view graph learning methods mostly focus on the multi-view consistency, but neglect the potential multi-view inconsistency (which may be incurred by noise, corruptions, or view-specific characteristics). To address this, this paper presents a new graph learning-based multi-view clustering approach, which for the first time, to our knowledge, simultaneously and explicitly formulates the multi-view *consistency* and the multi-view *inconsistency* in a unified optimization model. To solve this model, a new alternating optimization scheme is designed, where the consistent and inconsistent parts of each single-view graph as well as the unified graph that fuses the consistent parts of all views can be iteratively learned. It is noteworthy that our multi-view graph learning model is applicable to both similarity graphs and dissimilarity graphs, leading to two graph fusion-based variants, namely, distance (dissimilarity) graph fusion and similarity graph fusion. Experiments on various multi-view datasets demonstrate the superiority of our approach. The MATLAB source code is available at https://github.com/youweiliang/ConsistentGraphLearning.**

***Index Terms*—Multi-view graph learning; Multi-view clustering; Graph fusion; Consistency; Inconsistency.**

I. Introduction

Multi-view data consist of features collected from multiple heterogeneous sources (or views). The multiple views of features can provide rich and complementary information for discovering the underlying cluster structure of data. It has been a popular research topic in recent years as to how to effectively and jointly exploit the features from multiple views and thus achieve robust clusterings for multi-view data.

In the literature, numerous (single-view) clustering methods have been developed [1], among which the graph-based methods are one of the most widely-studied categories [2], [3]. The graph-based methods typically construct a similarity graph, and then partition this graph to obtain the clustering result. In these methods, the construction of the graph is independent of the clustering process, and the clustering performance heavily relies on the predefined graph. To alleviate this limitation, some (single-view) graph learning methods have be presented [4], [5], where the graph structure can be adaptively learned in the clustering process. More recently, inspired by the single-view graph learning [4], [5], the multi-view graph learning has rapidly emerged as a powerful technique for enhancing the multi-view clustering performance [6]–[9]. Remarkably, Zhan et al. [6]–[8] developed several multi-view graph learning approaches, which are able to fuse multiple graphs into a consistent graph with a certain number of connected components. Nie et al. [9] proposed a self-weighted scheme for fusing multiple graphs with the importance of each view considered. Despite the significant progress, a common limitation to these multi-view graph learning methods [6]–[9] lies in that they mostly focus on the consistency of multiple views, but lack the ability to explicitly consider both multi-view consistency and inconsistency (which may be brought in by noise, corruptions, or view-specific characteristics) in their frameworks, which can degrade their performances when faced with potentially noisy or low-quality data.

In the single-view scenario, to deal with the potential noise or corruptions, Bojchevski et al. [10] proposed a new graph-based clustering method based on the latent decomposition of the similarity graph into two graphs, namely, the *good* graph and the *corrupted* graph. Though it is able to learn a *good* graph by eliminating the influence of the potential noise, this graph learning method [10] is only applicable to a single graph (for a single view) and cannot be utilized in the multi-view graph learning task where multiple graphs from multiple views are involved. Thereby, it is still a very challenging problem how to jointly model the multi-view consistency (which can be viewed as the multi-view good graphs) as well as the multi-view inconsistency (which can be viewed as the multi-view corrupted graphs) in a unified multi-view graph learning model for improving the multi-view clustering performance.

To tackle this problem, this paper proposes a novel multi-view graph learning approach for multi-view clustering. We argue that the simultaneous modeling of multi-view consistency and multi-view inconsistency can significantly benefit the multi-view graph learning process. In particular, with the graph structures of multiple views given, their consistency and inconsistency are simultaneously leveraged to learn a unified graph. It is intuitive to assume that the graph of each view can be decomposed into two parts, i.e., the consistent part and the inconsistent part, and the goal is to learn and remove the inconsistent parts while preserving the consistent parts. Specifically, we formulate the multi-view consistency and multi-view inconsistency as well as the graph fusion term into a new objective function. By iteratively optimizing

*Corresponding author.

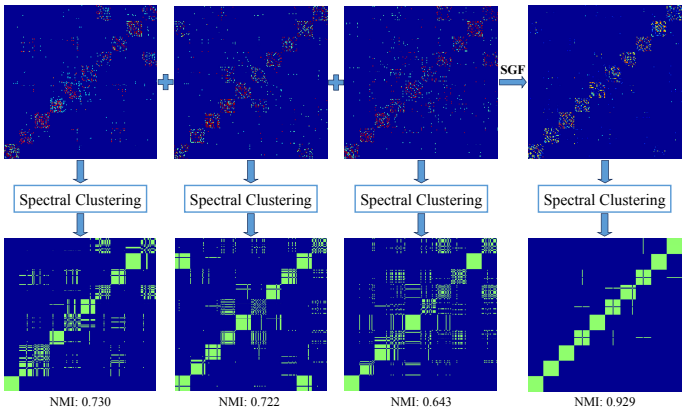


Fig. 1. Visualization of the similarity matrices on the UCI Digits dataset. Three views are used to test the proposed similarity graph fusion (SGF) algorithm. The first row corresponds to the three single-view similarity matrices and the fused similarity matrix (i.e., the fused graph). The second row corresponds to the clustering results by performing spectral clustering on the single-view graphs and the fused graph, respectively.

this objective function, the multi-view graph decomposition and the multi-view graph fusion are simultaneously achieved. With the fused graph obtained, some conventional graph-based methods like spectral clustering can be performed to obtain the final multi-view clustering result.

For clarity, we provide a visual example for our multi-view graph learning model in Figure 1. As shown in the first row of Figure 1, the three similarity matrices from three views appear to be corrupted to different extents, and our proposed similarity graph fusion (SGF) method is able to effectively remove many of these corruptions (or inconsistency) while yielding a unified and better graph with their consistent parts fused. As shown in the second row of Figure 1, by graph fusion with both consistency and inconsistency considered, the final clustering (in the fourth column) on the fused graph is significantly better than the clusterings on the original graphs.

The main contributions of this work are summarized below.

- We propose a new multi-view graph learning approach, which for the first time, to the best of our knowledge, simultaneously and explicitly models multi-view consistency and inconsistency in a unified objective function.
- To optimize this objective function, we present an efficient alternating minimization scheme to achieve an approximate solution.
- A novel multi-view clustering framework based on multi-view graph learning is presented, which is further extended into two graph fusion variants, namely, distance (dissimilarity) graph fusion and similarity graph fusion.

The rest of this paper is organized as follows. In Section II, we describe the proposed multi-view graph learning model. In Section III, we present an efficient algorithm to solve the optimization problem. In Section IV, two graph fusion versions for multi-view spectral clustering are proposed based on the framework. Finally, we report the experimental results in Section V and conclude this paper in Section VI.

II. Learning a Consistent Graph with Inconsistency Considered

In this section, we propose a new multi-view graph learning method which is capable of simultaneously and explicitly modeling multi-view consistency and inconsistency in a unified optimization framework.

Let $\mathbf{W}^{(i)} \in \mathbb{R}^{n\times n}_{\geqslant 0}$ denote the similarity matrix for the i-th view, with n being the number of instances (data points). The similarity matrices for different views may be significantly different even when they yield similar clustering results. For example, this is the case when the similarity matrix for i-th view is a multiple of the similarity matrix for j-th view, i.e., $\mathbf{W}^{(i)} = a \cdot \mathbf{W}^{(j)}$. Let $\mathbf{L}^{(i)}, \mathbf{L}^{(j)}$ be their (normalized) Laplacian matrices respectively. For all $k \in [1, n]$, the eigenvectors corresponding to the k-th largest eigenvalue of $\mathbf{L}^{(i)}$ and $\mathbf{L}^{(j)}$ are parallel. Thus they will give exactly the same clusters in normalized cut [2]. Therefore, we need to scale the similarity matrices before combining them into one common similarity matrix, i.e., multiply $\mathbf{W}^{(i)}$ by a scaling coefficient α_i. To make the scaling result unique, we restrict the sum of the coefficients to 1, i.e., $\boldsymbol{\alpha}^\top \cdot \mathbf{1} = 1$ in matrix form. All the scaled similarity matrices should be close to the common similarity matrix $\mathbf{S}$. Hence we want to minimize the following objective function with the constraints:

$$\begin{aligned} \min_{\boldsymbol{\alpha},\mathbf{S}} \quad & \sum_{i=1}^{v} \|\alpha_i \mathbf{W}^{(i)} - \mathbf{S}\|_F^2 \\ \text{s.t.} \quad & \boldsymbol{\alpha}^\top \mathbf{1} = 1, \alpha \geqslant 0, \mathbf{S} \geqslant 0. \end{aligned} \tag{1}$$

Here, v is the number of views.

To simultaneously model multi-view consistency and multi-view inconsistency, we decompose the similarity matrix $\mathbf{W}^{(i)}$ for the i-th view into two parts: the consistent part $\mathbf{A}^{(i)}$ and the inconsistent part $\mathbf{E}^{(i)}$. More formally, we assume that

$$\mathbf{W}^{(i)} = \mathbf{A}^{(i)} + \mathbf{E}^{(i)} \tag{2}$$

with $\mathbf{A}^{(i)}, \mathbf{E}^{(i)} \in \mathbb{R}^{n\times n}_{\geqslant 0}$. The key question here is how to find the matrices $\mathbf{A}^{(i)}$ and $\mathbf{E}^{(i)}$.

Some previous studies [10], [11] have modeled a similar decomposition, but they mainly focus on modeling the noise in the data. In contrast, this paper focuses on the concept of inconsistency. Although the noise is generally considered to be sparse on a similarity graph [10], [11], the inconsistency is no longer required to be.

The *inconsistency* can be seen as a much broader concept than *noise*. It may be caused by not just noise (or corruptions), but also different kinds of view-specific characteristics. Due to the diversity of different views, the inconsistency can appear everywhere on the similarity graphs. Thereby, the sparsity *within a similarity matrix* is no longer a necessary (or suitable) assumption in detecting the inconsistency on the multi-view similarity graphs. When decomposing a similarity matrix $\mathbf{W}^{(i)}$ into $\mathbf{A}^{(i)}$ and $\mathbf{E}^{(i)}$, the inconsistent part $\mathbf{E}^{(i)}$ is not necessary to be a sparse matrix. Instead, we argue that a more reasonable assumption is that the inconsistency should be sparse *across views*, i.e., the inconsistent parts from different views should

have little in common with each other. It is intuitive that the inconsistent parts are supposed to be inconsistent with each other, otherwise it would contradict the definition of inconsistency. To ensure that the inconsistency is sparse across views, we should make the sum of the products of the inconsistent parts to be small, that is

$$\gamma \sum_{\substack{i,j=1 \\ i\neq j}}^{V} \operatorname{sum}\left((\alpha_i \mathbf{E}^{(i)}) \circ (\alpha_j \mathbf{E}^{(j)})\right), \tag{3}$$

where $\circ$ denotes the element-wise multiplication of two matrices, and $\operatorname{sum}(\cdot)$ the operator of summing all elements in a matrix, and γ is a parameter. We scale the inconsistent part of each similarity matrices to address the aforementioned scale problem. In addition, we typically do not want the inconsistent parts to be too large, which is

$$\beta \sum_{i=1}^{V} \operatorname{sum}\left((\alpha_i \mathbf{E}^{(i)}) \circ (\alpha_i \mathbf{E}^{(i)})\right), \tag{4}$$

with β being a parameter.

To simultaneously model multi-view consistency and inconsistency in a unified optimization framework, we combine Eq. (1) (3) (4) into an overall objective function. Using the fact that $\operatorname{sum}\left((\alpha_i \mathbf{E}^{(i)}) \circ (\alpha_j \mathbf{E}^{(j)})\right) = \alpha_i \alpha_j \operatorname{Tr}\left(\mathbf{E}^{(i)} \cdot (\mathbf{E}^{(j)})^\top\right)$, where $\operatorname{Tr}(\cdot)$ denotes the matrix trace operator, we have the final optimization problem

$$\begin{aligned} \min_{\substack{\boldsymbol{\alpha},\mathbf{S}, \\ \mathbf{A}^{(1)},\ldots,\mathbf{A}^{(v)}}} \quad & \sum_{i=1}^{v} \left\|\alpha_i \mathbf{A}^{(i)} - \mathbf{S}\right\|_F^2 + \\ & \sum_{i,j=1}^{v} b_{ij}\alpha_i\alpha_j \operatorname{Tr}\left((\mathbf{W}^{(i)} - \mathbf{A}^{(i)}) \cdot (\mathbf{W}^{(j)} - \mathbf{A}^{(j)})^\top\right) \\ \text{s.t.} \quad & \boldsymbol{\alpha}^\top \mathbf{1} = 1, \boldsymbol{\alpha} \geqslant 0, \mathbf{S} \geqslant 0, \\ & \mathbf{W}^{(i)} \geqslant \mathbf{A}^{(i)} \geqslant 0, \quad i = 1, \ldots, v, \end{aligned} \tag{5}$$

where $\mathbf{B}$ is a v-by-v matrix with diagonal elements and non-diagonal elements being β and γ respectively. We do *not* require $\mathbf{S}$ to be symmetric, because we can set $\mathbf{S} = (\mathbf{S}+\mathbf{S}^\top)/2$ to make it symmetric after solving the optimization problem, just like we make the similarity matrix of k-nearest neighbor graph symmetric by setting $\mathbf{W} = (\mathbf{W} + \mathbf{W}^\top)/2$.

Why this objective can detect multi-view inconsistency? In ideal case, if all similarity matrices are consistent, the optimal value of the objective Eq. (1) would be 0. However, due to the inconsistency across views, it will not be 0. The higher the inconsistency, the larger the objective value. If the inconsistent parts are moved from the original similarity matrix $\mathbf{W}^{(i)}$ to the matrix $\mathbf{E}^{(i)}$, the first term Eq. (1) can be reduced to a smaller number, and the third term Eq. (4) will not increase much, as we can set β to be a small number; so does the second term Eq. (3), because inconsistency is sparse across views. Hence, the overall objective value will decrease. Therefore, the optimization process is actually shifting the inconsistent parts from the original similarity matrix $\mathbf{W}^{(i)}$ to the matrix $\mathbf{E}^{(i)}$ by minimizing the overall objective function. This idea is the core principle of how we simultaneously model multi-view consistency and multi-view inconsistency in a unified optimization framework.

III. Optimization

It can be proved that the constraint $\mathbf{S} \geqslant 0$ in Problem (5) can be removed while the global minimizer(s) remains the same, but the proof is omitted here due to the limitation of space. As the objective function is not jointly convex on all variables, we use the alternating minimization scheme to optimize it, i.e., we first optimize the objective function over $\boldsymbol{\alpha}, \mathbf{S}$ with $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(v)}$ fixed, and then optimize it over $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(v)}$ with $\boldsymbol{\alpha}, \mathbf{S}$ fixed, and repeat these two steps until the objective value converges. Specifically, we develop an efficient algorithm based on projection to solve these two sub-problems. Its outline is: 1) ignore the constraints and solve the unconstrained problem; 2) project the solution obtained in step 1 to the feasible region $\mathcal{G}$ (see below for the details).

1) Fix $\mathbf{A}^{(1)}, \ldots, \mathbf{A}^{(v)}$, update $\boldsymbol{\alpha}$ and $\mathbf{S}$: With $\mathbf{A}^{(i)}$ fixed and the inequality constraints removed (we keep the equation constraint $\boldsymbol{\alpha}^\top \mathbf{1} = 1$, since it is easy to solve), we have the following problem

$$\begin{aligned} \min_{\boldsymbol{\alpha},\mathbf{S}} \quad & \sum_{i=1}^{v} \left\|\alpha_i \mathbf{A}^{(i)} - \mathbf{S}\right\|_F^2 \\ & + \sum_{i,j=1}^{v} b_{ij}\alpha_i\alpha_j \operatorname{Tr}\left((\mathbf{W}^{(i)} - \mathbf{A}^{(i)}) \cdot (\mathbf{W}^{(j)} - \mathbf{A}^{(j)})^\top\right), \end{aligned} \tag{6}$$

$$\text{s.t.} \quad \boldsymbol{\alpha}^\top \mathbf{1} = 1. \tag{7}$$

Its Lagrangian function is

$$\begin{aligned} \mathcal{L}(\boldsymbol{\alpha}, \mathbf{S}, \mu) = & \sum_{i=1}^{v} \left\|\alpha_i \mathbf{A}^{(i)} - \mathbf{S}\right\|_F^2 \\ & + \sum_{i,j=1}^{v} b_{ij}\alpha_i\alpha_j \operatorname{Tr}\left((\mathbf{W}^{(i)} - \mathbf{A}^{(i)}) \cdot (\mathbf{W}^{(j)} - \mathbf{A}^{(j)})^\top\right) \\ & + \mu(\boldsymbol{\alpha}^\top \mathbf{1} - 1), \end{aligned} \tag{8}$$

where μ is the Lagrange multiplier. The first order necessary conditions for optimality are

$$\begin{aligned} & \frac{\partial \mathcal{L}}{\partial \alpha_i} = -2 \operatorname{Tr}\left(\mathbf{A}^{(i)}\mathbf{S}^\top\right) + 2\alpha_i \operatorname{Tr}\left(\mathbf{A}^{(i)}(\mathbf{A}^{(i)})^\top\right) + \\ & 2\sum_{j=1}^{v} b_{ij}\alpha_j \operatorname{Tr}\left((\mathbf{W}^{(i)} - \mathbf{A}^{(i)}) \cdot (\mathbf{W}^{(j)} - \mathbf{A}^{(j)})^\top\right) + \mu = 0, \\ & i = 1, \ldots, v, \end{aligned} \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{S}} = 2v\mathbf{S} - 2\sum_{i=1}^{v} \alpha_i \mathbf{A}^{(i)} = \mathbf{0}, \tag{10}$$

$$\boldsymbol{\alpha}^\top \mathbf{1} = 1. \tag{11}$$

From Eq. (10) we have

$$\mathbf{S} = \frac{1}{v}\sum_{i=1}^{v} \alpha_i \mathbf{A}^{(i)} \tag{12}$$

Substitute Eq. (12) into Eq. (9). We have

$$2\sum_{j=1}^{v}\alpha_j\left(-\frac{1}{v}\operatorname{Tr}(\mathbf{A}^{(i)}(\mathbf{A}^{(j)})^\top)\right)+$$
$$2\sum_{j=1}^{v}\alpha_j b_{ij}\operatorname{Tr}\left((\mathbf{W}^{(i)}-\mathbf{A}^{(i)})\cdot(\mathbf{W}^{(j)}-\mathbf{A}^{(j)})^\top\right)+$$
$$2\alpha_i\operatorname{Tr}(\mathbf{A}^{(i)}(\mathbf{A}^{(i)})^\top)+\mu=0,\quad i=1,\ldots,v. \tag{13}$$

Let $f_{ij}=b_{ij}\cdot\operatorname{Tr}((\mathbf{W}^{(i)}-\mathbf{A}^{(i)})\cdot(\mathbf{W}^{(j)}-\mathbf{A}^{(j)})^\top)$, $g_{ij}=\operatorname{Tr}(\mathbf{A}^{(i)}(\mathbf{A}^{(j)})^\top)$, and let $\mathbf{G}=(g_{ij})_{v\times v}$, $\mathbf{F}=(f_{ij})_{v\times v}$. Then Eq. (13) becomes

$$2((\mathbf{e}_i-\frac{1}{v}\mathbf{1})\circ\mathbf{g}_i+\mathbf{f}_i)^\top\boldsymbol{\alpha}+\mu=0,\quad i=1,\ldots,v, \tag{14}$$

which can be written in matrix form

$$2((\mathbf{I}-\frac{1}{v}\mathbb{1})\circ\mathbf{G}+\mathbf{F})\cdot\boldsymbol{\alpha}+\mu\mathbf{1}=\mathbf{0}, \tag{15}$$

where $\mathbf{1}$ is a v-by-1 vector with all components equal to 1, and $\mathbf{e}_i$ is a v-by-1 vector with all components equal to 0, except the i-th, which is 1. $\mathbf{I}\in\mathbb{R}^{v\times v}$ is a v-by-v identity matrix, and $\mathbb{1}$ is a v-by-v matrix with all elements equal 1. Combine Eq. (7) and Eq. (15), and let

$$\mathbf{H}=2((\mathbf{I}-\frac{1}{v}\mathbb{1})\circ\mathbf{G}+\mathbf{F}),$$

we have the following system of linear equations of $[\boldsymbol{\alpha},\mu]^\top$:

$$\begin{bmatrix}\mathbf{H} & \mathbf{1}\\ \mathbf{1}^\top & 0\end{bmatrix}\cdot\begin{bmatrix}\boldsymbol{\alpha}\\ \mu\end{bmatrix}=\begin{bmatrix}\mathbf{0}\\ 1\end{bmatrix}. \tag{16}$$

We can obtain the solution $\boldsymbol{\alpha}$ using basic linear algebra, and then project it onto the feasible region $\mathcal{G}_0=\{\boldsymbol{\alpha}\geqslant 0:\boldsymbol{\alpha}^\top\mathbf{1}=1\}$ by solving the following problem.

$$\min_{\boldsymbol{\alpha}}\quad\|\boldsymbol{\alpha}-\tilde{\boldsymbol{\alpha}}\|_2^2 \tag{17}$$
$$\text{s.t.}\quad\boldsymbol{\alpha}\geqslant 0,\boldsymbol{\alpha}^\top\mathbf{1}=1. \tag{18}$$

This is a projection problem on the probability simplex, which can be solved by [12].

2) Fix $\boldsymbol{\alpha}$ and $\mathbf{S}$, update $\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(v)}$: With $\boldsymbol{\alpha}$ and $\mathbf{S}$ fixed and constraints removed, we have the following problem

$$\min_{\mathbf{A}^{(1)},\ldots,\mathbf{A}^{(v)}}\quad\sum_{i=1}^{v}\left\|\alpha_i\mathbf{A}^{(i)}-\mathbf{S}\right\|_F^2+$$
$$\sum_{i,j=1}^{v}b_{ij}\alpha_i\alpha_j\operatorname{Tr}\left((\mathbf{W}^{(i)}-\mathbf{A}^{(i)})\cdot(\mathbf{W}^{(j)}-\mathbf{A}^{(j)})^\top\right). \tag{19}$$

Set the derivative of Eq. (19) to 0 and rearrange it, we have

$$\alpha_i^2\mathbf{A}^{(i)}+\sum_{j=1}^{v}b_{ij}\alpha_i\alpha_j\mathbf{A}^{(j)}=\alpha_i\mathbf{S}+\sum_{j=1}^{v}b_{ij}\alpha_i\alpha_j\mathbf{W}^{(j)},$$
$$i=1,\ldots,v, \tag{20}$$

which are v matrix equations. Taking a closer look at them, we find that they can be transformed into n^2 systems of linear equations of $[a_{ij}^{(1)},\ldots,a_{ij}^{(v)}]^\top$ with the same coefficient matrix:

$$\mathbf{C}=(\mathbf{B}+\mathbf{I})\circ(\boldsymbol{\alpha}\boldsymbol{\alpha}^\top) \tag{21}$$

Denote $\alpha_i\mathbf{S}+\sum_{j=1}^{v}b_{ij}\alpha_i\alpha_j\mathbf{W}^{(j)}$ by $\mathbf{H}^{(i)}$, and let $\mathbf{h}^{(i)}=\operatorname{vec}(\mathbf{H}^{(i)})$, where $\operatorname{vec}(\cdot)$ is the vectorization operator. The systems of linear equations are described by

$$\mathbf{C}\cdot\begin{bmatrix}\operatorname{vec}(\mathbf{A}^{(1)})\\ \vdots\\ \operatorname{vec}(\mathbf{A}^{(v)})\end{bmatrix}=\begin{bmatrix}\mathbf{h}^{(1)}\\ \vdots\\ \mathbf{h}^{(v)}\end{bmatrix} \tag{22}$$

Its solution is given by

$$\begin{bmatrix}\operatorname{vec}(\mathbf{A}^{(1)}\\ \vdots\\ \operatorname{vec}(\mathbf{A}^{(v)})\end{bmatrix}=\mathbf{C}^+\cdot\begin{bmatrix}\mathbf{h}^{(1)}\\ \vdots\\ \mathbf{h}^{(i)}\end{bmatrix}, \tag{23}$$

where $\mathbf{C}^+$ denotes the pseudo-inverse of $\mathbf{C}$. After getting $\operatorname{vec}(\mathbf{A}^{(i)})$, we can obtain $\mathbf{A}^{(i)}$ by reshaping $\operatorname{vec}(\mathbf{A}^{(i)})$ into a matrix. To project $\mathbf{A}^{(i)}$ onto the feasible region $\mathcal{G}_i=\{\mathbf{A}^{(i)}:\mathbf{W}^{(i)}\geqslant\mathbf{A}^{(i)}\geqslant 0\}$, we simply set

$$\mathbf{A}^{(i)}=\max(\mathbf{A}^{(i)},0),\mathbf{A}^{(i)}=\min(\mathbf{A}^{(i)},\mathbf{W}^{(i)}), \tag{24}$$

where $\max()$ and $\min()$ are applied element-wise.

For clarity, the overall algorithm of consistent graph learning is summarized in Algorithm 1.

Algorithm 1 Consistent Graph Learning

Input: Adjacency matrices $\{\mathbf{W}^{(1)},\ldots,\mathbf{W}^{(v)}\}$, parameters β and γ, max iteration m
Output: Adjacency matrix of the consistent graph $\mathbf{S}$

```
Initialize A^(i): A^(i) ← W^(i),  i = 1,...,v
while not converge do
    Obtain α̃ by solving Eq. (16)
    Project α̃ onto the feasible region using Alg. [12]
    Update S using Eq. (12)
    Obtain vec(A^(i)) by Eq. (23)
    Reshape vec(A^(i)) into a matrix A^(i)
    Project A^(i) onto the feasible region using Eq. (24)
    if reach max iteration then
        break
    end if
end while
```

IV. Two Graph Fusion Versions

By applying our multi-view graph learning model to similarity graphs and distance (dissimilarity) graphs, respectively, we further propose two specific algorithms, namely, similarity graph fusion (**SGF**) and distance graph fusion (**DGF**).

In **SGF**, we use v similarity graphs as input for Algorithm 1. For each view, the (Euclidean) distance is first transformed to similarity by Gaussian kernel, and a k-nearest neighbor (kNN) similarity graph is then built. Slightly different from the classic kNN graph, we will keep the edge between two points x_i and x_j as long as x_i is among the kNNs of x_j in *any* view. With the v similarity graphs fused into a consistent similarity graph, spectral clustering is then used to obtain the final clustering.

In **DGF**, we use v distance graphs as input. For each view, we first build a kNN distance graph by the (Euclidean) distance. The distance graphs are fused into a unified dissimilarity

TABLE I
STATISTICS OF THE REAL-WORLD DATASETS

dataset	# of instances	# of views	# of clusters
UCI Digits	2000	6	10
NUS-WIDE	2000	5	31
MSRCv1	210	5	7
Flower17	1360	7	17
Caltech101-7	1474	6	7
Caltech101-20	2386	6	20
BBCSport	544	2	5
Reuters	1500	5	6

graph by Algorithm 1. Then, the unified dissimilarity graph is transformed into a similarity graph by Gaussian kernel, upon which spectral clustering is used to obtain the final result.

V. EXPERIMENT

In this section, we compare the two proposed graph learning based multi-view clustering algorithms, namely, **SGF** and **DGF**, against seven state-of-the-art multi-view spectral clustering algorithms, namely, co-regularized spectral clustering (**CoRegSC**) [13], robust multi-view spectral clustering (**RMSC**) [11], affinity aggregation for spectral clustering (**AASC**) [14], weighted multi-view spectral clustering based on spectral perturbation (**WMSC**) [15], multiview clustering via adaptively weighted procrustes (**AWP**) [16], graph learning for multiview clustering (**GLMC**) [6], and multiview consensus graph clustering (**MCGC**) [8]. Besides these multi-view algorithms, the classic spectral clustering (**SC**) [2] is also performed on each view of the datasets, and the best single-view performance by SC is reported for reference only.

A. Datasets and Evaluation Metric

We conduct experiments on eight real-world multi-view datasets, namely, UCI Handwritten Digits [17], NUS-WIDE [18], MSRCv1 [19], Flower17 [20], Caltech101-7 [21], and Caltech101-20 [21] for images clustering, Reuters [17], and BBCSport [22] for news article clustering. Due to the large size of the original NUS-WIDE and Reuters datasets, we randomly sample two subsets from them in our experiments. The details of the datasets are given in Table I.

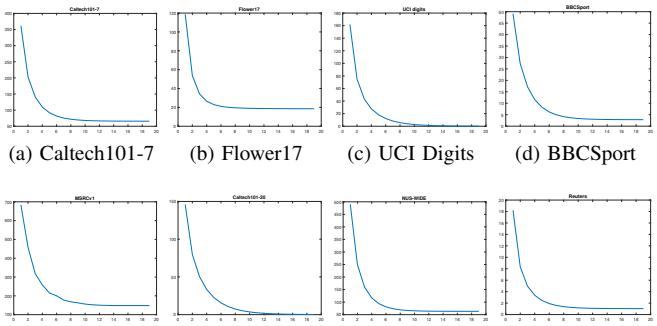
In the experiments, we use the normalized mutual information (NMI) [21] as the evaluation metric.

B. Experiments Setup

For each algorithm, the grid search is used to search the parameter(s) from 10^{-6} to 10^{6} on logarithmic scale and its performance with the best parameter(s) is reported. We run all the algorithms 20 times, and report their average scores and standard deviations.

In our **DGF** method, we use Euclidean distance for all datasets except the two text datasets, i.e., the BBCSport and Reuters datasets, where the cosine distance is used.

Similarly, in **SGF**, we use the cosine similarity to build similarity matrices for the BBCSport and Reuters datasets and use Gaussian kernels for other datasets.



(a) Caltech101-7 (b) Flower17 (c) UCI Digits (d) BBCSport

(e) MSRCv1 (f) Caltech101-20 (g) NUS-WIDE (h) Reuters

Fig. 2. Convergence curves of the proposed algorithm on the eight datasets. The Y-axis are the objective value, and the X-axis are number of iterations.

C. Results and Analysis

We report the clustering performances of different multi-view clustering methods in Table II. As can be seen in Table II, the two proposed graph learning methods, i.e., **SGF** and **DGF**, achieve better performances than the state-of-the-art methods on most of the datasets, which demonstrate the robustness of our algorithms. Note that **AASC**, **GLMC**, and **MCGC** are also graph learning (or graph fusion) based methods, which typically focus on multi-view consistency yet cannot (explicitly) incorporate the multi-view *inconsistency* in their models. The experimental results in Table II have shown the advantages of our algorithms over **AASC**, **GLMC**, and **MCGC**, probably due to our simultaneous modeling of multi-view consistency and multi-view inconsistency in our framework.

It is noteworthy that our distance (dissimilarity) graph fusion version generally performs better than our similarity graph fusion version. As previous graph learning methods mostly exploit similarity graphs, this comparative study can serve as an interesting start for researchers to consider the shift from *similarity graph learning* to *dissimilarity graph learning* for multi-view clustering.

D. Convergence Analysis

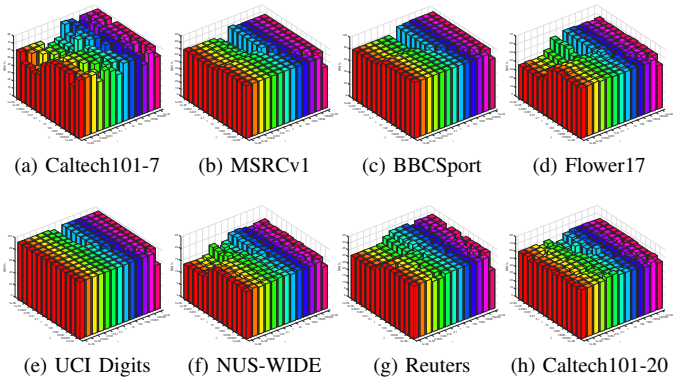
Because the original function is not jointly convex on all variables, we may not obtain a global minimum. We propose an alternating minimization scheme to solve the optimization problem. Although the projection-based method does not guarantee to converge, it is reliable and converges within a few iterations in practice, as shown in Figure 2.

E. Parameters Sensitivity

We have two parameters β and γ in the proposed algorithm. As showed in Figure 3, we test each parameter from 10^{-6} to 10^{6} on log scale while fixing the value of the other parameter. The results indicate that the performance of the proposed algorithm is stable across a wide range of parameters. Note that we typically do not need to tune these two parameters, as we can obtain quite stable performance by simply setting β in the range $[10^{-6}, 10^{-4}]$ and γ in the range $[10^{4}, 10^{6}]$ across different datasets, as Figure 3 indicates.

TABLE II
AVERAGE PERFORMANCES (W.R.T. NMI (%)) OVER 20 RUNS BY DIFFERENT ALGORITHMS. THE BEST TWO SCORES IN EACH COLUMN ARE IN BOLD.

Method	Caltech101-7	MSRCv1	BBCSport	Flower17	UCI Digits	NUS-WIDE	Reuters	Caltech101-20
SC(best)	$52.42_{\pm 0.97}$	$62.46_{\pm 0.00}$	$81.73_{\pm 0.00}$	$46.94_{\pm 0.42}$	$84.69_{\pm 0.04}$	$17.27_{\pm 0.25}$	$30.32_{\pm 0.04}$	$54.36_{\pm 0.97}$
CoRegSC	$48.21_{\pm 0.00}$	$75.89_{\pm 0.15}$	$\mathbf{93.15}_{\pm 0.00}$	$55.50_{\pm 0.13}$	$93.65_{\pm 0.04}$	$19.20_{\pm 0.27}$	$\mathbf{36.93}_{\pm 0.00}$	$56.79_{\pm 1.06}$
RMSC	$49.45_{\pm 0.14}$	$73.97_{\pm 0.00}$	$91.63_{\pm 3.38}$	$55.57_{\pm 0.35}$	$85.96_{\pm 1.10}$	$19.10_{\pm 0.27}$	$34.38_{\pm 0.00}$	$59.88_{\pm 1.03}$
AASC	$53.85_{\pm 0.07}$	$75.12_{\pm 0.51}$	$90.34_{\pm 0.00}$	$57.98_{\pm 0.21}$	$88.64_{\pm 0.03}$	$19.58_{\pm 0.26}$	$33.44_{\pm 0.00}$	$61.35_{\pm 0.48}$
MVGL	$55.52_{\pm 0.00}$	$70.86_{\pm 0.00}$	$92.47_{\pm 0.00}$	$45.50_{\pm 0.00}$	$88.91_{\pm 0.00}$	$10.32_{\pm 0.00}$	$27.62_{\pm 0.00}$	$59.07_{\pm 0.00}$
MCGC	$51.26_{\pm 0.00}$	$69.62_{\pm 0.00}$	$91.42_{\pm 0.00}$	$50.43_{\pm 0.64}$	$94.22_{\pm 0.00}$	$16.31_{\pm 0.53}$	$30.10_{\pm 0.00}$	$59.59_{\pm 0.00}$
AWP	$48.59_{\pm 1.44}$	$68.98_{\pm 4.32}$	$89.84_{\pm 6.99}$	$51.49_{\pm 1.20}$	$88.65_{\pm 4.18}$	$17.15_{\pm 0.42}$	$30.61_{\pm 2.89}$	$56.86_{\pm 1.75}$
WMSC	$51.22_{\pm 0.00}$	$75.34_{\pm 0.34}$	$92.85_{\pm 0.00}$	$57.93_{\pm 0.50}$	$91.04_{\pm 0.04}$	$19.04_{\pm 0.30}$	$35.02_{\pm 0.70}$	$57.48_{\pm 0.81}$
SGF	$\mathbf{56.07}_{\pm 0.06}$	$\mathbf{76.92}_{\pm 0.14}$	$92.28_{\pm 0.00}$	$\mathbf{64.83}_{\pm 0.21}$	$\mathbf{94.54}_{\pm 0.00}$	$\mathbf{19.61}_{\pm 0.40}$	$35.04_{\pm 0.03}$	$\mathbf{61.58}_{\pm 0.72}$
DGF	$\mathbf{75.55}_{\pm 5.02}$	$\mathbf{81.29}_{\pm 0.00}$	$\mathbf{94.05}_{\pm 0.00}$	$\mathbf{58.13}_{\pm 0.53}$	$\mathbf{96.22}_{\pm 0.00}$	$\mathbf{19.93}_{\pm 0.28}$	$\mathbf{39.52}_{\pm 0.80}$	$\mathbf{65.36}_{\pm 0.92}$



(a) Caltech101-7 (b) MSRCv1 (c) BBCSport (d) Flower17

(e) UCI Digits (f) NUS-WIDE (g) Reuters (h) Caltech101-20

Fig. 3. NMI against parameters β and γ on each dataset.

VI. CONCLUSIONS

This paper presents a new graph learning-based multi-view clustering approach, which simultaneously and explicitly formulates multi-view consistency and multi-view inconsistency in a unified optimization model. To solve this model, we present an efficient optimization algorithm which combines alternating minimization scheme with projection method to obtain an approximate solution. Further, the proposed approach is extended to two graph fusion versions, corresponding to distance (dissimilarity) graph fusion and similarity graph fusion, respectively. Experimental results have shown the superiority of our approach against several state-of-the-art multi-view spectral clustering approaches on eight multi-view datasets.

It is noteworthy that our distance graph fusion version generally performs better than our similarity graph fusion version on the benchmark datasets. A probable reason for this finding is that the fusion of distance matrices (before the kernel function) may better preserve the structure information than the fusion of similarity matrices (after the kernel function), as the kernel function can bring some bias into the graph when mapping the distance into a similarity. Based on the theoretical and empirical evidence of this paper, it would be an interesting direction to conduct more in-depth investigation on the *consistency VS inconsistency* issue as well as the *similarity fusion VS dissimilarity fusion* issue in the future multi-view graph learning research.

ACKNOWLEDGMENT

This work was supported by NSFC (61976097, 61602189, and 61876193).

REFERENCES

[1] A. K. Jain, "Data clustering: 50 years beyond k-means," *PRL*, vol. 31, no. 8, pp. 651–666, 2010.

[2] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and Computing*, vol. 17, no. 4, pp. 395–416, 2007.

[3] D. Huang, C.-D. Wang, J.-S. Wu, J.-H. Lai, and C.-K. Kwoh, "Ultra-scalable spectral clustering and ensemble clustering," *IEEE TKDE*, 2019.

[4] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *KDD*, 2014, pp. 977–986.

[5] F. Nie, X. Wang, M. I. Jordan, and H. Huang, "The constrained laplacian rank algorithm for graph-based clustering," in *AAAI*, 2016.

[6] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE TCYB*, vol. 48, no. 10, pp. 2887–2895, 2018.

[7] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, and Y. Yang, "Graph structure fusion for multiview clustering," *IEEE TKDE*, 2018.

[8] K. Zhan, F. Nie, J. Wang, and Y. Yang, "Multiview consensus graph clustering," *IEEE TIP*, vol. 28, no. 3, pp. 1261–1270, 2019.

[9] F. Nie, J. Li, and X. Li, "Self-weighted multiview clustering with multiple graphs." in *IJCAI*, 2017, pp. 2564–2570.

[10] A. Bojchevski, Y. Matkovic, and S. Günnemann, "Robust spectral clustering for noisy data: Modeling sparse corruptions improves latent embeddings," in *KDD*, 2017, pp. 737–746.

[11] R. Xia, Y. Pan, L. Du, and J. Yin, "Robust multi-view spectral clustering via low-rank and sparse decomposition," in *AAAI*, 2014.

[12] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1-ball for learning in high dimensions," in *ICML*, 2008, pp. 272–279.

[13] A. Kumar, P. Rai, and H. Daumé, "Co-regularized multi-view spectral clustering," in *NeurIPS*, 2011, pp. 1413–1421.

[14] H.-C. Huang, Y.-Y. Chuang, and C.-S. Chen, "Affinity aggregation for spectral clustering," in *CVPR*, 2012, pp. 773–780.

[15] L. Zong, X. Zhang, X. Liu, and H. Yu, "Weighted multi-view spectral clustering based on spectral perturbation," in *AAAI*, 2018.

[16] F. Nie, L. Tian, and X. Li, "Multiview clustering via adaptively weighted procrustes," in *KDD*, 2018, pp. 2022–2030.

[17] D. Dua and C. Graff, "UCI machine learning repository," 2017. [Online]. Available: http://archive.ics.uci.edu/ml

[18] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y.-T. Zheng, "NUS-WIDE: A real-world web image database from national university of singapore," in *CIVR*, 2009.

[19] J. Winn and N. Jojic, "Locus: Learning object classes with unsupervised segmentation," in *ICCV*, vol. 1, 2005, pp. 756–763.

[20] M.-E. Nilsback and A. Zisserman, "A visual vocabulary for flower classification," in *CVPR*, vol. 2, 2006, pp. 1447–1454.

[21] Y. Li, F. Nie, H. Huang, and J. Huang, "Large-scale multi-view spectral clustering via bipartite graph," in *AAAI*, 2015.

[22] D. Greene and P. Cunningham, "Practical solutions to the problem of diagonal dominance in kernel document clustering," in *ICML*, 2006, pp. 377–384.