Learnable Graph Filter for Multi-view Clustering

Peng Zhou
Anhui University
Hefei, China
zhoupeng@ahu.edu.cn

Liang Du*
Shanxi University
Taiyuan, China
duliang@sxu.edu.cn

ABSTRACT

Multi-view clustering is an important machine learning task for multi-media data. Recently, graph filter based multi-view clustering achieves promising performance and attracts much attention. However, the conventional graph filter based methods only use a pre-defined graph filter for each view and the used graph filters ignore the rich information among all views. Different from the conventional methods, in this paper, we aim to tackle a new problem, i.e., instead of using the pre-defined graph filters, how to construct an appropriate consensus graph filter by considering the information in all views. To achieve this, we propose a novel multi-view clustering method with graph filter learning. In our method, we learn an appropriate consensus graph filter from all views of data with multiple graph learning rather than directly pre-defining it. Then, we provide an iterative algorithm to obtain the consensus graph filter and analyze why it can lead to better clustering results. The extensive experiments on benchmark data sets demonstrate the effectiveness and superiority of the proposed method. The codes of this article are released in http://Doctor-Nobody.github.io/codes/MCLGF.zip.

CCS CONCEPTS

• **Computing methodologies → Machine learning algorithms**.

KEYWORDS

Multi-view clustering, graph filter learning, multiple graph learning

ACM Reference Format:
Peng Zhou and Liang Du. 2023. Learnable Graph Filter for Multi-view Clustering. In *Proceedings of the 31st ACM International Conference on Multimedia (MM '23), October 29–November 3, 2023, Ottawa, ON, Canada.* ACM, New York, NY, USA, 10 pages. https://doi.org/10.1145/3581783.3611912

1 INTRODUCTION

In real-world multi-media applications, many data are represented in multiple views, which are called multi-view data. For example, a web page may contain several views of content such as texts, images, and videos. To handle these multi-view data, multi-view learning is proposed and becomes an important field of research in the multi-media and machine learning community [9, 17, 37, 38, 48–51]. Among them, multi-view clustering attracts increasingly more attention because it does not need any annotations or labels, which makes it more easily used in real-world applications.

Multi-view clustering adopts the consensus and complementary information among multiple views to learn a consensus clustering result. For example, Kumar et al. learned the consensus result by applying the co-regularized term in multi-view spectral clustering [15]; Huang et al. designed non-linear fusion method for multi-view clustering with self-paced learning [11]; Wen et al. discovered the consensus and complementary information in the graphs of all views and proposed a multi-view clustering method to handle incomplete multi-view data [35]. Among them, graph filter based multi-view clustering is one of the new and promising methods [10, 12, 18]. These methods first obtain more cluster-friendly representations of multi-views with the graph filters and then learn a consensus result on these cluster-friendly representations.

Although the graph filter based methods achieve promising performance, they still have two limitations. Firstly, their graph filters are often *pre-defined or designed manually.* As we know, the effect of the graph filter depends on the quality of the corresponding graph. However, unfortunately, it is difficult to tell which graph is appropriate for a given data or a given view in advance. The pre-defined graph filters constructed from an inappropriate graph may not improve or even deteriorate the clustering performance. Secondly, the previous works design the filters *for each view independently*, which means the filters cannot adopt the rich consensus and complementary information among different views. Therefore, to further improve the performance, we should be more careful to design the graph filters for multi-view clustering.

To address these issues, in this paper, instead of directly applying the graph filter to do multi-view clustering, we focus on an alternative question, i.e., how to learn an appropriate graph filter for multi-view clustering. To this end, we propose a novel Multi-view Clustering method based on Learnable Graph Filter (MCLGF). Different from conventional methods which construct the graph filter for each view independently, we aim to learn one consensus graph filter for all views so that the filter may consider the consensus and complementary information among all views. To achieve this, we learn the appropriate graph filter in a multiple graph learning framework, which can effectively ensemble the information in all views. Although the introduced objective function seems complicated, we provide an ADMM method [2] which can effectively optimize it to learn the consensus graph filter. We also provide some theoretical analysis to show that with the learned graph filter, we can indeed obtain a more cluster-friendly representation. The extensive experiments on multi-view data show that the proposed method can outperform the compared multi-view clustering methods and even the state-of-the-art graph filter based methods.

[*] Corresponding author.

MM '23, October 29–November 3, 2023, Ottawa, ON, Canada.

ACM ISBN 979-8-4007-0108-5/23/10…$15.00
https://doi.org/10.1145/3581783.3611912

We summarize the main contributions of this paper as follows:

- Different from conventional methods, which only directly apply the graph filter to improve the clustering performance, we focus on answering a new question which is how to learn an appropriate graph filter for multi-view data.
- We propose a novel multi-view clustering method with a learnable graph filter via multiple graph learning. With multiple graph learning, the learned graph filter can effectively adopt rich information in all views.
- We conduct extensive experiments on benchmark data sets to demonstrate the effectiveness of the proposed multi-view clustering method.

2 RELATED WORK AND PRELIMINARIES

In this section, we briefly introduce some related works and preliminaries of the multi-view clustering and graph filter.

2.1 Multi-view Clustering

Multi-view clustering often learns a consensus clustering result from multiple views by extending some single-view clustering methods with considering the consensus and complementary in all views. For example, Cai et al. extended the kmeans from single view to multi-view data leading to the multi-view kmeans method [3]; Liu et al. proposed the multi-view non-negative matrix factorization method for multi-view data [7].

Since spectral clustering is one of the most famous clustering methods, many works extend spectral clustering to the multi-view setting. For example, Xia et al. provided a robust multi-view spectral clustering method with low-rank and sparse decomposition [36]; Nie et al. proposed some parameter-free and auto-weighted multi-view spectral clustering methods [24–26]; Tao et al. designed some robust multi-view spectral clustering methods with ensemble clustering [30, 31]; Zhou et al. proposed an incremental multi-view spectral clustering method which can handle the data with large number of views [52]; Li et al. applied the spectral clustering on the consensus graph learned from the multiple views [16]; Zong et al. designed the multi-view spectral clustering based on the spectral perturbation [53].

Another popular clustering method is subspace clustering and thus many multi-view subspace clustering methods are proposed. For example, Zhang et al. learned the latent representations of the data for multi-view subspace clustering [41, 42]; Kang et al. designed a large scale multi-view subspace clustering method in linear time with bipartite graph [13]; Zhao et al. presented a robust multi-view subspace clustering method by learning the consensus representation [45]; Zhang et al. designed a one-step multi-view subspace clustering method without any postprocessing [44]; Zhang et al. proposed a multi-view subspace clustering method by considering the low-rank structure [43].

This paper proposes a spectral-based multi-view clustering method by learning an appropriate consensus graph filter.

2.2 Graph Filter

Considering an undirected weighted graph $\mathcal{G}$ with n vertices $\{v_1, \cdots, v_n\}$, its adjacency matrix is $\mathbf{W} \in \mathbb{R}^{n\times n}$ where $\mathbf{W}$ is symmetric and $W_{ij} \geq 0$. Then, we can construct its normalized Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-\frac{1}{2}}\mathbf{W}\mathbf{D}^{-\frac{1}{2}}$, where $\mathbf{I}$ is an identity matrix and $\mathbf{D}$ is a diagonal matrix whose diagonal elements are the summation of the rows of $\mathbf{W}$. Consider the eigenvalue decomposition of $\mathbf{L}$: $\mathbf{L} = \mathbf{U}\Sigma\mathbf{U}^T$, where $\mathbf{U} \in \mathbb{R}^{n\times n}$ is composed of n eigenvectors of $\mathbf{L}$ and $\Sigma \in \mathbb{R}^{n\times n}$ is a diagonal matrix whose diagonal elements are n eigenvalues of $\mathbf{L}$. From the perspective of spectral graph theory, the eigenvectors of $\mathbf{L}$ are the Fourier bases of the graph and the eigenvalues are the associated frequencies [28].

Now, given a graph signal $\mathbf{s} = [s(v_1), \cdots, s(v_n)]^T$ on the graph $\mathcal{G}$, the graph filter is a transform or an operation $\mathcal{F}$ on the graph signal $\mathbf{s}$. In the clustering task, the data feature matrix $\mathbf{X} \in \mathbb{R}^{n\times d}$ with n instances and d features can be regarded as d graph signals. If data $\mathbf{X}$ has a clearer clustering structure, it should follow the *cluster and manifold assumption*, which is that the data in the same cluster should be close to each other.

Previous works [23, 27] show that smoother signals $\mathbf{X}$ will have a clearer clustering structure which follows the cluster and manifold assumption. Therefore, to obtain better clustering performance, we should use a graph filter on the signals $\mathbf{X}$ to make it smoother. According to [18, 23], smooth signals should contain more low-frequency bases than high-frequency bases. Therefore, one popular graph filter is defined as :

$$\mathcal{F}(\mathbf{s}) = \mathbf{U}\left(\mathbf{I} - \frac{\Sigma}{2}\right)^r \mathbf{U}^T\mathbf{s} = \left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^r \mathbf{s}, \tag{1}$$

where r is a positive integer to capture the r-hop neighborhood high-order relation. Notice that small eigenvalues in $\mathbf{L}$, which corresponds to the low-frequency parts, lead to large eigenvalues in $\left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^r$, and thus the graph filter can preserve the low-frequency parts and suppress the high-frequency parts. Here we use $\left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^r$ instead of directly use $(\mathbf{I} - \mathbf{L})^r$ because all the eigenvalues in $\mathbf{L}$ are in the range $[0, 2]$. By the filter $\left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^r$, the transformed eigenvalues are in the range $[0, 1]$. If we use the filter $(\mathbf{I} - \mathbf{L})^r$, the transformed eigenvalues may be smaller than 0.

Since the graph filter can lead to a smoother signal which is a kind of cluster-friendly embedding of the original data, it is applied in multi-view clustering and obtains promising performance. For example, Ma et al. and Huang et al. applied the graph filter to multi-view subspace clustering method [10, 23]; Pan et al. used it in the multi-view clustering with contrastive graph learning [27]; Lin et al. and Hang et al. proposed the graph filter based multi-view attributed graph clustering [12, 18].

As introduced before, the above-mentioned methods often use an independent pre-defined graph filter for each view to transform each view of the data to a cluster-friendly embedding and learn the consensus clustering result from the multiple embeddings. In this paper, we aim to learn a more appropriate consensus graph filter by considering the information of all views, which can further improve the clustering performance.

3 METHOD

We first introduce some notations. We use boldface uppercase letters to denote the matrices and use boldface lowercase letters to denote the vectors. Given a matrix $\mathbf{M}$, we use $M_{i.}$ and $M_{.i}$ to denote its i-th row vector and column vector, respectively. We use M_{ij} to denote its (i, j)-th element.

Given a multi-view data set $\mathcal{X} = \{\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(m)}\}$ with m views, $\mathbf{X}^{(v)} \in \mathbb{R}^{n \times d_v}$ is the v-th view of $\mathcal{X}$, where n is the number of instances and d_v is the number of features in the v-th view. Then we can construct the k-nn graph for each view. In more detail, taking the v-th view as example, we first compute its similar matrix $\mathrm{S}^{(v)} \in \mathbb{R}^{n \times n}$ with heat kernel as follows:

$$S_{ij}^{(v)} = e^{-\frac{\|\mathrm{X}_{i.}^{(v)} - \mathrm{X}_{j.}^{(v)}\|_2^2}{2\sigma^2}}, \tag{2}$$

where σ is the bandwidth parameter and we set it as the median of the Euclidean distances of all pairs. Then we construct the k-nn graph $\mathcal{G}^{(v)}$ whose adjacency matrix is $\mathbf{W}^{(v)}$ from $\mathrm{S}^{(v)}$. If $\mathbf{X}_{i.}^{(v)}$ is one of the k neighbors of $\mathbf{X}_{j.}^{(v)}$ or $\mathbf{X}_{j.}^{(v)}$ is one of the k neighbors of $\mathbf{X}_{i.}^{(v)}$, then $W_{ij}^{(v)} = S_{ij}^{(v)}$, or otherwise $W_{ij}^{(v)} = 0$. Specially, we set the diagonal elements of $\mathbf{W}^{(v)}$ to all 1s. In our implementation, we fix the numbers of neighbors $k = 10$ for simplicity. Obviously, $\mathbf{W}^{(v)}$ is symmetric and non-negative. After constructing multiple graphs from multiple views, we can learn a consensus graph for the graph filter.

3.1 Multiple Graph Learning

After obtaining $\mathbf{W}^{(1)}, \cdots, \mathbf{W}^{(m)}$, we aim to learn a consensus graph $\mathcal{G}$ whose adjacency matrix is $\mathbf{W} \in \mathbb{R}^{n \times n}$. Since the quality of each view differs, we impose a weight $0 \leq \alpha_v \leq 1$ for each view and wish the better view has a larger weight. Then, we can obtain the following loss function:

$$\begin{aligned} \min_{\mathbf{W}, \boldsymbol{\alpha}} \quad & \sum_{v=1}^{m} \alpha_v^2 \|\mathbf{W} - \mathbf{W}^{(v)}\|_F^2, \\ s.t. \quad & \mathbf{W} = \mathbf{W}^T, \quad 0 \leq W_{ij} \leq 1, \quad \sum_{j=1}^{n} W_{ij} = 1, \\ & 0 \leq \alpha_v \leq 1, \quad \sum_{v=1}^{m} \alpha_v = 1, \end{aligned} \tag{3}$$

where the first constraint on $\mathbf{W}$ ensures that the adjacency matrix is symmetric, and the second constraint makes the adjacency matrix bounded and non-negative. The third constraint on $\mathbf{W}$ is a row normalization to make the summation of each row to be 1. Since this constraint works like an ℓ_1 norm on each row of $\mathbf{W}$, it can make the learned graph more sparse. With the consensus graph $\mathcal{G}$, we can learn a consensus graph filter for all views.

3.2 Graph Filter Learning

As introduced before, the graph filter can lead to a cluster-friendly embedding of the data, and thus we will learn an appropriate filter on $\mathcal{X}$ to make the data more easily for clustering. Since we aim to learn a consensus graph filter for all views, we will learn the filter with consensus $\mathbf{W}$ used in Eq.(3). Given $\mathbf{W}$, its normalized Laplacian is $\mathbf{L} = \mathbf{I} - \mathbf{D}^{-1/2}\mathbf{W}\mathbf{D}^{-1/2} = \mathbf{I} - \mathbf{W}$ because $\mathbf{D} = \mathbf{I}$ as shown in the third constraint on $\mathbf{W}$ in Eq.(3). Then the graph filter can be defined as $\left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^r = \left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^r$.

Given the v-th view, we regard the feature matrix of the v-th view $\mathbf{X}^{(v)}$ as the graph signals, and then we operate the above-mentioned graph filter on the signals $\mathbf{X}^{(v)}$ to obtain a more cluster-friendly embedding $\mathcal{F}(\mathbf{X}^{(v)})$ as follows:

$$\mathcal{F}(\mathbf{X}^{(v)}) = \left(\frac{\mathbf{I} + \mathbf{W}}{2}\right)^r \mathbf{X}^{(v)}. \tag{4}$$

Here, the i-th row of $\mathcal{F}(\mathbf{X}^{(v)})$ (which we denote as $\mathcal{F}(\mathbf{X}^{(v)})_{i.}$) is the embedding of the i-th instance in the v-th view.

With these embeddings, we can construct the objective function to learn an appropriate adjacency matrix $\mathbf{W}$ for the graph filter. We wish that the embeddings $\mathcal{F}(\mathbf{X}^{(v)})$ can preserve the manifold structure of the v-th view. In more detail, given i-th and j-th instances $\mathbf{X}_{i.}^{(v)}$ and $\mathbf{X}_{j.}^{(v)}$ in the v-th view, if $W_{ij}^{(v)}$ is large, which means in the v-th view, $\mathbf{X}_{i.}^{(v)}$ and $\mathbf{X}_{j.}^{(v)}$ are similar, then we wish the embeddings $\mathcal{F}(\mathbf{X}^{(v)})_{i.}$ and $\mathcal{F}(\mathbf{X}^{(v)})_{j.}$ also be similar. It can be achieved by minimizing $\frac{1}{2}\sum_{i,j=1}^{n} W_{ij}^{(v)} \|\mathcal{F}(\mathbf{X}^{(v)})_{i.} - \mathcal{F}(\mathbf{X}^{(v)})_{j.}\|_2^2$. Taking the definition of $\mathcal{F}(\cdot)$ (i.e., Eq.(4)) into it, we obtain the following objective function:

$$\min_{\mathbf{W}} \frac{1}{2} \sum_{v=1}^{m} \sum_{i,j=1}^{n} W_{ij}^{(v)} \left\| \left(\left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^r \mathbf{X}^{(v)} \right)_{i.} - \left(\left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^r \mathbf{X}^{(v)} \right)_{j.} \right\|_2^2. \tag{5}$$

Combining Eq.(3) and Eq.(5), we obtain our final objective function:

$$\begin{aligned} \min_{\mathbf{W}, \boldsymbol{\alpha}} \quad & \frac{1}{2} \sum_{v=1}^{m} \sum_{i,j=1}^{n} W_{ij}^{(v)} \left\| \left(\left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^r \mathbf{X}^{(v)} \right)_{i.} - \left(\left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^r \mathbf{X}^{(v)} \right)_{j.} \right\|_2^2 \\ & + \lambda \sum_{v=1}^{m} \alpha_v^2 \|\mathbf{W} - \mathbf{W}^{(v)}\|_F^2, \\ s.t. \quad & \mathbf{W} = \mathbf{W}^T, \quad 0 \leq W_{ij} \leq 1, \quad \sum_{j=1}^{n} W_{ij} = 1, \\ & 0 \leq \alpha_v \leq 1, \quad \sum_{v=1}^{m} \alpha_v = 1, \end{aligned} \tag{6}$$

where λ is a balanced hyper-parameter. Notice that, different from the conventional graph filter based multi-view clustering methods, which operate a pre-defined graph filter on each view of the data matrix independently to obtain the embedding and do the multi-view clustering on the embedding, our formula Eq.(6) focuses on learning an appropriate consensus graph filter for all views. Since the graph filter in our method is learned from all views of data, it can be more appropriate for the multi-view clustering task.

3.3 Optimization

Before optimizing Eq.(6), we first reformulate it to make it more easily for optimization. By expanding the first term in Eq.(6), we can rewrite it as:

$$\begin{aligned} & \frac{1}{2} \sum_{i,j=1}^{n} W_{ij}^{(v)} \left\| \left(\left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^r \mathbf{X}^{(v)} \right)_{i.} - \left(\left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^r \mathbf{X}^{(v)} \right)_{j.} \right\|_2^2 \\ =& tr\left(\mathbf{X}^{(v)T} \left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^{rT} (\mathbf{D}^{(v)} - \mathbf{W}^{(v)}) \left(\frac{\mathbf{I}+\mathbf{W}}{2}\right)^{r} \mathbf{X}^{(v)} \right) \end{aligned} \tag{7}$$

where $\mathbf{D}^{(v)}$ is a diagonal matrix whose diagonal elements $D_{ii}^{(v)} = \sum_{j=1}^{n} W_{ij}^{(v)}$. Then we can take Eq.(7) into Eq.(6). However, since the objective function and the constraints w.r.t. $\mathbf{W}$ are very complicated, we apply ADMM [2] to optimize it. In more detail, we first introduce two auxiliary variables $\mathbf{B} = \frac{\mathbf{I}+\mathbf{W}}{2}$ and $\mathbf{V} = \mathbf{W}$, and obtain the following equivalent objective function:

$$\begin{aligned}
\min_{\mathbf{W},\boldsymbol{\alpha},\mathbf{B},\mathbf{V}} \; & tr\left(\mathbf{X}^{(v)T}\mathbf{B}^{rT}(\mathbf{D}^{(v)} - \mathbf{W}^{(v)})\mathbf{B}^{r}\mathbf{X}^{(v)}\right) + \lambda\sum_{v=1}^{m}\alpha_v^2\|\mathbf{W} - \mathbf{W}^{(v)}\|_F^2, \\
s.t. \quad & \mathbf{V} = \mathbf{W}, \quad 0 \le W_{ij} \le 1, \quad \sum_{j=1}^{n} W_{ij} = 1, \\
& \mathbf{B} = \frac{\mathbf{I}+\mathbf{W}}{2}, \quad \mathbf{V} = \mathbf{V}^T, \quad 0 \le \alpha_v \le 1, \quad \sum_{v=1}^{m}\alpha_v = 1. \qquad (8)
\end{aligned}$$

Then, we can obtain its Lagrange formula by introducing the Lagrange multipliers $\boldsymbol{\Lambda}_1 \in \mathbb{R}^{n\times n}$ and $\boldsymbol{\Lambda}_2 \in \mathbb{R}^{n\times n}$:

$$\begin{aligned}
\mathcal{L} =& tr\left(\mathbf{X}^{(v)T}\mathbf{B}^{rT}(\mathbf{D}^{(v)} - \mathbf{W}^{(v)})\mathbf{B}^{r}\mathbf{X}^{(v)}\right) + \lambda\sum_{v=1}^{m}\alpha_v^2\|\mathbf{W} - \mathbf{W}^{(v)}\|_F^2 \\
&+ tr\left(\boldsymbol{\Lambda}_1^T\left(\mathbf{B} - \frac{\mathbf{I}+\mathbf{W}}{2}\right)\right) + tr\left(\boldsymbol{\Lambda}_2^T\left(\mathbf{W} - \mathbf{V}\right)\right) \qquad (9) \\
&+ \frac{\mu}{2}\left(\left\|\mathbf{B} - \frac{\mathbf{I}+\mathbf{W}}{2}\right\|_F^2 + \|\mathbf{W} - \mathbf{V}\|_F^2\right)
\end{aligned}$$

where $\mu > 0$ is an adaptive parameter. Now, we optimize $\mathbf{B}, \mathbf{W}, \mathbf{V}$, and $\boldsymbol{\alpha}$ iteratively by fixing other variables.

3.3.1 Optimizing $\mathbf{B}$. The subproblem w.r.t. $\mathbf{B}$ can be written as:

$$\begin{aligned}
\min_{\mathbf{B}} \quad \mathcal{J} =& tr\left(\mathbf{X}^{(v)T}\mathbf{B}^{rT}(\mathbf{D}^{(v)} - \mathbf{W}^{(v)})\mathbf{B}^{r}\mathbf{X}^{(v)}\right) \qquad (10) \\
&+ tr\left(\boldsymbol{\Lambda}_1^T\left(\mathbf{B} - \frac{\mathbf{I}+\mathbf{W}}{2}\right)\right) + \frac{\mu}{2}\left\|\mathbf{B} - \frac{\mathbf{I}+\mathbf{W}}{2}\right\|_F^2
\end{aligned}$$

Notice that Eq.(10) is a non-constraint optimization problem, which can be solved by the standard Quasi-Newton method. In our implementation, we use L-BFGS algorithm [20] to optimize it. To apply the L-BFGS algorithm, we need the partial derivative of $\mathcal{J}$ w.r.t. $\mathbf{B}$. According to the chain rule of the derivative, we have

$$\begin{aligned}
\frac{\partial\mathcal{J}}{\partial\mathbf{B}} =& \sum_{t=0}^{r-1} 2\mathbf{B}^{tT}(\mathbf{D}^{(v)} - \mathbf{W}^{(v)})\mathbf{X}^{(v)}\mathbf{X}^{(v)T}(\mathbf{B}^{r-1-t})^T \\
&+ \boldsymbol{\Lambda}_1 + \frac{\mu}{2}\left(\mathbf{B} - \frac{\mathbf{I}+\mathbf{W}}{2}\right). \qquad (11)
\end{aligned}$$

Then we can take it into the L-BFGS algorithm to obtain the solution of $\mathbf{B}$.

3.3.2 Optimizing $\mathbf{W}$. When optimizing $\mathbf{W}$, we can reformulate the objective function as the following form:

$$\begin{aligned}
\min_{\mathbf{W}} \quad & \|\mathbf{W} - \mathbf{A}\|_F^2, \qquad (12) \\
s.t. \quad & 0 \le W_{ij} \le 1, \quad \sum_{j=1}^{n} W_{ij} = 1,
\end{aligned}$$

where $\mathbf{A} = \frac{\lambda\sum_{v=1}^{m}\alpha_v^2\mathbf{W}^{(v)} + \frac{\boldsymbol{\Lambda}_1}{4} - \frac{\boldsymbol{\Lambda}_2}{2} + \frac{\mu}{4}\left(\mathbf{B} - \frac{\mathbf{I}}{2} + 2\mathbf{V}\right)}{\lambda\sum_{v=1}^{m}\alpha_v^2 + \frac{5\mu}{8}}$.

Eq.(12) can be decoupled into n independent subproblems by rows. Therefore, we solve Eq.(12) row by row. Considering the i-th row of Eq.(12), it is a problem of Euclidean projection onto the simplex, whose closed-form solution can be obtained by a standard method such as [5].

3.3.3 Optimizing $\mathbf{V}$. The subproblem w.r.t. $\mathbf{V}$ is as follows:

$$\begin{aligned}
\min_{\mathbf{V}} \quad & \left\|\mathbf{V} - \left(\mathbf{W} + \frac{\boldsymbol{\Lambda}_2}{\mu}\right)\right\|_F^2, \qquad (13) \\
s.t. \quad & \mathbf{V} = \mathbf{V}^T.
\end{aligned}$$

It is also a Euclidean projection problem, whose closed-form solution is:

$$\mathbf{V} = \frac{\mathbf{W} + \mathbf{W}^T}{2} + \frac{\boldsymbol{\Lambda}_2 + \boldsymbol{\Lambda}_2^T}{2\mu} \qquad (14)$$

3.3.4 Optimizing $\boldsymbol{\alpha}$. When fixing other variables, we obtain the following formula:

$$\begin{aligned}
\min_{\boldsymbol{\alpha}} \quad & \sum_{v=1}^{m}\alpha_v^2\|\mathbf{W} - \mathbf{W}^{(v)}\|_F^2, \qquad (15) \\
s.t. \quad & 0 \le \alpha_v \le 1, \quad \sum_{v=1}^{m}\alpha_v = 1.
\end{aligned}$$

According to the Cauchy-Schwarz Inequality, its closed-form solution is:

$$\alpha_v = \frac{\|\mathbf{W} - \mathbf{W}^{(v)}\|_F^{-2}}{\sum_{v=1}^{m}\|\mathbf{W} - \mathbf{W}^{(v)}\|_F^{-2}}. \qquad (16)$$

Notice that $\|\mathbf{W} - \mathbf{W}^{(v)}\|_F^2$ indicates the difference between the graph of the v-th view and the graph of the consensus view. If a view is far away from the consensus one, which means its quality is low, since $\alpha_v \propto 1/\|\mathbf{W} - \mathbf{W}^{(v)}\|_F^2$, its α_v will be small, which means the weight of the low-quality view will be small. It is consistent with our motivation for the weights.

3.3.5 Updating the Lagrange Multipliers. We update the Lagrange multipliers $\boldsymbol{\Lambda}_1$, $\boldsymbol{\Lambda}_2$, and the parameter μ as following:

$$\begin{cases}
\boldsymbol{\Lambda}_1 \leftarrow \boldsymbol{\Lambda}_1 + \mu\left(\mathbf{B} - \frac{\mathbf{I}+\mathbf{W}}{2}\right), \\
\boldsymbol{\Lambda}_2 \leftarrow \boldsymbol{\Lambda}_2 + \mu(\mathbf{W} - \mathbf{V}), \\
\mu \leftarrow 1.05 * \mu.
\end{cases} \qquad (17)$$

After iteratively solving these variables, we obtain the final clustering result by running spectral clustering on the consensus matrix $\mathbf{W}$. The whole process is summarized in Algorithm 1.

3.4 Theoretical Analysis

In this subsection, we discuss why the learned graph filter can improve the clustering performance theoretically. According to [14], the low eigenvalues of a graph Laplacian matrix correspond to the large-scale structure, such as clusters, and the high eigenvalues correspond to the details and noises. Therefore, to obtain a clearer clustering structure, we need the *low-pass graph filter*, which can suppress the high eigenvalues and preserve the low ones. Consider a graph whose Laplacian matrix is $\mathbf{L}$ with n eigenvalues $0 = \sigma_1 \le \sigma_2 \le \cdots \le \sigma_n$. Let $\mathcal{H}(\cdot) : \mathbb{R}^{n\times n} \to \mathbb{R}^{n\times n}$ be a transform on the Laplacian matrix and the eigenvalues of $\mathcal{H}(\mathbf{L})$ are $h(\sigma_1), \cdots, h(\sigma_n)$,

Algorithm 1 Multi-view Clustering method based on Learnable Graph Filter

```
Input: Multi-view data $\mathcal{X} = \{\mathbf{X}^{(1)}, \cdots, \mathbf{X}^{(m)}\}$, hyper-parameter $r$ and $\lambda$.
Output: Final consensus clustering result.
Construct $\mathbf{W}^{(v)}$ for each view by Eq.(2).
Initialize $\mathbf{W} = \frac{1}{m}\sum_{v=1}^{m}\mathbf{W}^{(v)}$, $\alpha = \frac{1}{m}$, $\mathbf{V} = \mathbf{W}$, $\mathbf{\Lambda}_1 = \mathbf{\Lambda}_2 = \mathbf{0}$, and $\mu = 1$.
while not converge do
    Update $\mathbf{B}$ by solving Eq.(10).
    Update $\mathbf{W}$ by solving Eq.(12).
    Update $\mathbf{V}$ by Eq.(14).
    Update $\boldsymbol{\alpha}$ by solving Eq.(16).
    Update Lagrange multipliers by Eq.(17).
end while
Obtain the final clustering result by applying spectral clustering on the consensus graph $\mathbf{W}$.
```

where $h(\cdot)$ is the transform of the eigenvalues. Wai et al. gave the following Definition of the low-pass graph filter [33]:

Definition 1. *[33] (Low-pass graph filter) $\mathcal{H}(\mathbf{L})$ is a (K, η) low-pass graph filter if*

$$\eta := \frac{\max(|h(\sigma_{K+1})|, |h(\sigma_{K+2})|, \cdots, |h(\sigma_n)|)}{\min(|h(\sigma_1)|, |h(\sigma_2)|, \cdots, |h(\sigma_K)|)} < 1 \tag{18}$$

η is the low-pass coefficient. Definition 1 shows that, given a graph filter $\mathcal{H}(\mathbf{L})$, if there exists an integer $1 \leq K < n$ and a coefficient $\eta < 1$, then $\mathcal{H}(\mathbf{L})$ is a low-pass graph filter. Now, we show that the learned filter is a low-pass graph filter with the following Theorem.

Theorem 1. *Given the learned graph filter $(\frac{\mathbf{I}+\mathbf{W}}{2})^r$ of Algorithm 1, there exists an integer $1 \leq K < n$, which makes the low-pass coefficient η as defined in Definition 1 be smaller than 1, and thus the learned graph filter is a low-pass graph filter.*

Proof. Given the learned adjacency matrix $\mathbf{W}$ by Algorithm 1, its normalized Laplacian matrix $\mathbf{L} = \mathbf{I} - \mathbf{W}$. Supposing the eigenvalues of $\mathbf{L}$ are $0 = \sigma_1 \leq \sigma_2 \leq \cdots \leq \sigma_n$, the eigenvalues of $(\frac{\mathbf{I}+\mathbf{W}}{2})^r$ are $(1 - \frac{\sigma_1}{2})^r, \cdots, (1 - \frac{\sigma_n}{2})^r$. Then, we compute its low-pass coefficient η defined in Definition 1:

$$\begin{aligned}\eta =& \frac{\max(|(1-\frac{\sigma_{K+1}}{2})^r|, |(1-\frac{\sigma_{K+2}}{2})^r|, \cdots, |(1-\frac{\sigma_n}{2})^r|)}{\min(|(1-\frac{\sigma_1}{2})^r|, |(1-\frac{\sigma_2}{2})^r|, \cdots, |(1-\frac{\sigma_K}{2})^r|)} \\ =& \frac{(1-\frac{\sigma_{K+1}}{2})^r}{(1-\frac{\sigma_K}{2})^r} = \left(\frac{2-\sigma_{K+1}}{2-\sigma_K}\right)^r.\end{aligned} \tag{19}$$

Since $\sigma_1 = 0$, as long as there exists a non-zero eigenvalue of $\mathbf{L}$, there exists a K such that $\sigma_K < \sigma_{K+1}$, and thus $\eta = \left(\frac{2-\sigma_{K+1}}{2-\sigma_K}\right)^r < 1$. Therefore, our learned graph filiter $(\frac{\mathbf{I}+\mathbf{W}}{2})^r$ is a low-pass graph filter according to Definition 1. □

Theorem 1 shows that our learned filter is a low-pass graph filter and thus can well reveal the clustering structure of the data.

Moreover, we can analyze the performance from the viewpoint of spectral graph theory. In the clustering task, if data matrix $\mathbf{X}$ has a clear clustering structure, it should follow the cluster and manifold assumption. According to [23, 27], the cluster and manifold assumption requires that the data or graph signals should be smooth. Here, we follow the metric of smoothness defined in [6, 47]:

Definition 2. *[6](Smoothness) Given a graph with the adjacency matrix $\mathbf{W} \in \mathbb{R}^{n\times n}$ whose Laplacian matrix is $\mathbf{L}$, and a graph signal $\mathbf{x} \in \mathbb{R}^n$, the smoothness of signal graph $\mathbf{x}$ is defined as:*

$$\mathbf{x}^T\mathbf{L}\mathbf{x} = \frac{1}{2}\sum_{i,j=1}^{n} W_{ij}(x_i - x_j)^2. \tag{20}$$

According to Definition 2, in the following Theorem, we will show that, given the data of any view $\mathbf{X}^{(v)}$, if we regard $\mathbf{X}^{(v)}$ as graph signals, the signals operated by our filter (i.e., $\mathcal{F}(\mathbf{X}^{(v)})$) are smoother than the original signals (i.e., $\mathbf{X}^{(v)}$).

Theorem 2. *Given the learned graph filter $(\frac{\mathbf{I}+\mathbf{W}}{2})^r$ of Algorithm 1, the new signals $\mathcal{F}(\mathbf{X}^{(v)}) = (\frac{\mathbf{I}+\mathbf{W}}{2})^r\mathbf{X}^{(v)}$ is smoother than the original signals $\mathbf{X}^{(v)}$.*

Proof. Since $\mathbf{X}^{(v)}$ and $\mathcal{F}(\mathbf{X}^{(v)})$ both are composed of d_v independent graph signals, we will consider the i-th signal (i.e., $\mathbf{X}^{(v)}_{.i}$ and $\mathcal{F}(\mathbf{X}^{(v)})_{.i}$) as an example, and the results of other signals are similar. According to Definition 2, to prove $\mathcal{F}(\mathbf{X}^{(v)})_{.i}$ is smoother than $\mathbf{X}^{(v)}_{.i}$, we should prove $\mathcal{F}(\mathbf{X}^{(v)})^T_{.i}\mathbf{L}\mathcal{F}(\mathbf{X}^{(v)})_{.i} \leq \mathbf{X}^{(v)T}_{.i}\mathbf{L}\mathbf{X}^{(v)}_{.i}$.

Given our learned adjacency matrix $\mathbf{W}$, we have $\mathbf{L} = \mathbf{I} - \mathbf{W}$ and thus $\frac{\mathbf{I}+\mathbf{W}}{2} = \mathbf{I} - \frac{\mathbf{L}}{2}$. Denote the eigenvalue decomposition of $\mathbf{L}$ is $\mathbf{L} = \mathbf{U}\Sigma\mathbf{U}^T$, where $\mathbf{U}$ contains the eigenvectors of $\mathbf{L}$ and Σ contains the eigenvalues $0 = \sigma_1 \leq \cdots, \leq \sigma_n$. Then we have

$$\begin{aligned}\mathcal{F}(\mathbf{X}^{(v)})^T_{.i}\mathbf{L}\mathcal{F}(\mathbf{X}^{(v)})_{.i} =& \mathbf{X}^T_{.i}\left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^r \mathbf{L}\left(\mathbf{I} - \frac{\mathbf{L}}{2}\right)^r \mathbf{X}_{.i} \\ =& (\mathbf{U}^T\mathbf{X}_{.i})^T\left(\mathbf{I} - \frac{\Sigma}{2}\right)^r \Sigma \left(\mathbf{I} - \frac{\Sigma}{2}\right)^r (\mathbf{U}^T\mathbf{X}_{.i}).\end{aligned}$$

Denoting $\mathbf{z} = \mathbf{U}^T\mathbf{X}_{.i}$, we obtain:

$$\begin{aligned}&\mathcal{F}(\mathbf{X}^{(v)})^T_{.i}\mathbf{L}\mathcal{F}(\mathbf{X}^{(v)})_{.i} - \mathbf{X}^{(v)T}_{.i}\mathbf{L}\mathbf{X}^{(v)}_{.i} \\ =& \mathbf{z}^T\left(\mathbf{I} - \frac{\Sigma}{2}\right)^r \Sigma \left(\mathbf{I} - \frac{\Sigma}{2}\right)^r \mathbf{z} - \mathbf{z}^T\Sigma\mathbf{z} \\ =& \sum_{i=1}^{n}\left(\left(1 - \frac{\sigma_i}{2}\right)^{2r} - 1\right)\sigma_i z_i^2 \leq 0,\end{aligned} \tag{21}$$

where the inequality holds because all the eigenvalues σ_is of Laplacian matrix $\mathbf{L}$ satisfy $0 \leq \sigma_i \leq 2$. It shows that $\mathcal{F}(\mathbf{X}^{(v)})^T_{.i}\mathbf{L}\mathcal{F}(\mathbf{X}^{(v)})_{.i} \leq \mathbf{X}^{(v)T}_{.i}\mathbf{L}\mathbf{X}^{(v)}_{.i}$ which means $\mathcal{F}(\mathbf{X}^{(v)})_{.i}$ is smoother than $\mathbf{X}^{(v)}_{.i}$. □

Theorem 2 shows that with the learned graph filter, we can smooth the graph signals and obtain a clearer clustering structure, which follows the cluster and manifold assumption,

At last, we analyze the time complexity of Algorithm 1. Algorithm 1 only involves the matrix multiplication operations, therefore we just need to analyze the matrix multiplication. Denote n as the number of instances and d as the number of features in the view which contains the most features. When optimizing $\mathbf{B}$, we need to compute the partial derivative Eq.(11). By applying the

Table 1: Description of the data sets.

	#instances	#features	#classes
3sources	169	3560, 3631, 3068	6
Caltech	9144	48, 40, 254, 1984, 512, 928	102
CCV	6773	20, 20, 20	20
CiteSeer	3312	3312, 3703	6
COIL	1440	1024, 944, 4096, 576	20
Hdigit	10000	784, 256	10
NUSWIDE	2000	64, 225, 144 73, 128	31
Reuters	1500	21531, 24892, 34251 15506, 11547	6
Scene	4485	20, 59, 40	15
SUNRGBD	10335	4096, 4096	45

associative property of matrix multiplication, we can compute the partial derivative in $O(n^2d)$ time. When optimizing $\mathbf{W}$, we solve it row by row. Considering the i-th view, we can solve the Euclidean projection on simplex in $O(nlogn)$ time. Since there are n rows in $\mathbf{W}$, it costs $O(n^2logn)$ time to optimize $\mathbf{W}$. Obviously, this step can be easily parallelized. Optimizing $\mathbf{V}$ and $\boldsymbol{\alpha}$ only involves matrix addition, which is often very fast. Therefore, the bottleneck of the time complexity is $O(n^2d + n^2logn)$. This is comparable with the mainstream graph based multi-view clustering methods. Despite this, in the future, we will study how to speed up it further.

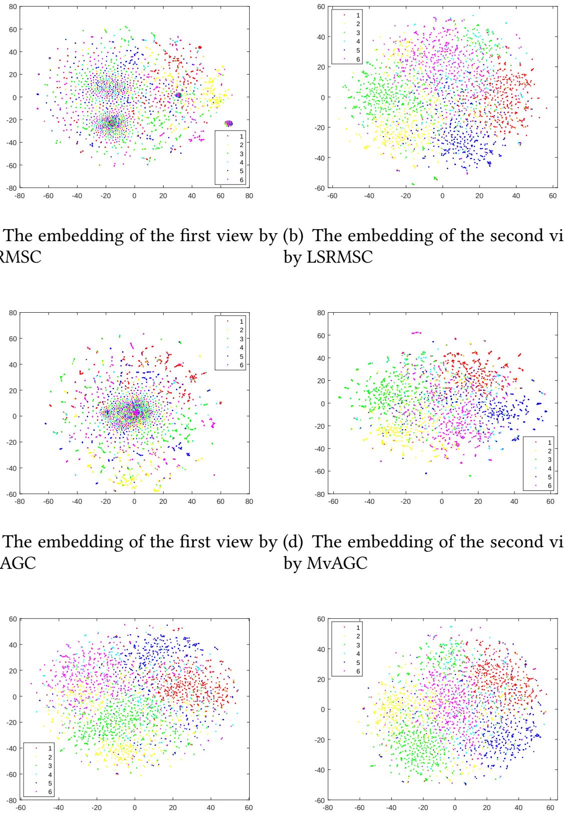
4 EXPERIMENTS

4.1 Data Sets

We conduct experiments on 10 benchmark data sets, including 3sources[1], Caltech[2], CCV[3], CiteSeer [8], COIL[4], Hdigit[5], NUSWIDE [4], Reuters [1], Scene[6] and SUNRGBD [46]. The detailed information of these data sets is shown in Table 1.

4.2 Experimental Setup

To show the effectiveness of the proposed method, we compare it with 16 state-of-the-art multi-view clustering methods, including RMSC [36], AMGL [25], MVGL [40], AWP [26], MCGC [39], CGD [29], GMC [34], LMVSC [13], 2CMV [22], LMSC [41], OPLFMVC [21], CGL [16], COMVSC [44], ONMVSC [17], LSRMSC [10], and MvAGC [19]. For all methods on all data sets, the number of clusters are set as the true number of classes. In our method, we fix $r = 2$ and tune λ in $[10^{-5}, 10^5]$. Two widely used metric Accuracy (ACC) and Normalized Mutual Information (NMI) are used to measure the clustering performance. The experiments are conducted using MATLAB on a PC with Windows 10, 4.2-GHz CPU, and 64-GB memory.



(a) The embedding of the first view by LSRMSC

(b) The embedding of the second view by LSRMSC

(c) The embedding of the first view by MvAGC

(d) The embedding of the second view by MvAGC

(e) The embedding of the first view by our MCLGF

(f) The embedding of the second view by our MCLGF

Figure 1: t-sne of the embedding of the two views in CiteSeer by LSRMSC, MvAGC, and the proposed MCLGF .

4.3 Experimental Results

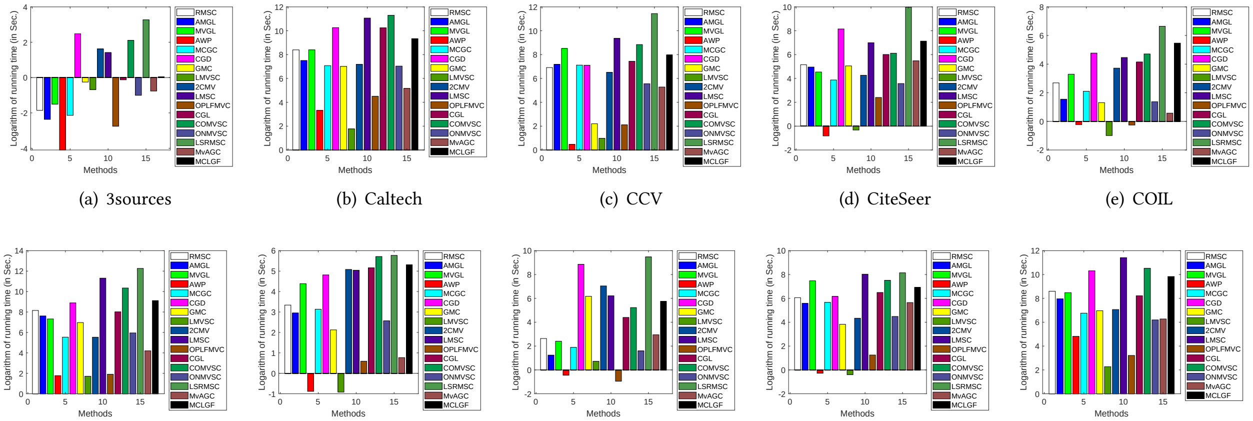
Tables 2 and 3 show the ACC and NMI results of all methods on all data sets. The red texts indicate the best results, the blue ones indicate the second best results, and the green ones indicate the third best results. Notice that, LSRMSC cannot run a result in reasonable time on the large data sets Caltech and SUNRGBD due to its high time complexity. From these Tables, we can find that the proposed MCLGF outperforms the state-of-the-art multi-view clustering methods on most data sets. On other data sets, MCLGF can still achieve comparable performance even though it is not the best one.

When comparing with other graph filter based methods LSRMSC and MvAGC, we show the t-sne [32] of the embeddings of the two views in CiteSeer data set by LSRMSC, MvAGC, and our MCLGF, respectively. The t-sne results are shown in Figure 1. We can see that, in the first view, our method can obtain a better embedding result because it can partition the data into different classes more clearly. In the center of Figure 1(a) and 1(c), LSRMSC and MvAGC both entangle data in different classes seriously, whereas our method can partition them well. In the second view, these methods obtain

[1] http://mlg.ucd.ie/datasets/3sources.html
[2] https://data.caltech.edu/records/mzrjq-6wc02
[3] https://www.ee.columbia.edu/ln/dvmm/CCV/
[4] https://www.cs.columbia.edu/CAVE/software/softlib/coil-20.php
[5] https://cs.nyu.edu/~roweis/data.html
[6] https://figshare.com/articles/dataset/15-Scene_Image_Dataset/7007177

Table 2: ACC results on all the data sets. Red texts indicate the best results, blue texts indicate the second best results, and green texts indicate the third best results.

Methods	3sources	Caltech	CCV	CiteSeer	COIL	Hdigit	NUSWIDE	Reuters	Scene	SUNRGBD
RMSC [36]	0.4260	0.1339	0.2578	0.2255	0.4660	0.3260	0.1645	0.3267	0.1507	0.1296
AMGL [25]	0.3254	0.2496	0.2317	0.2141	0.8417	0.8485	0.1565	0.2940	0.3271	0.2107
MVGL [40]	0.3077	0.1418	0.1124	0.2189	0.8090	0.9958	0.1450	0.3173	0.1889	0.1233
AWP [26]	0.4260	0.2613	0.1680	0.2554	0.6986	0.7239	0.1515	0.3247	0.3663	0.1723
MCGC [39]	0.3491	0.2090	0.1062	0.2120	0.8069	0.1002	0.1490	0.3327	0.1474	0.1823
CGD [29]	0.7870	0.2418	0.1540	0.3312	0.7660	0.7139	0.1485	0.4767	0.4230	0.2137
GMC [34]	0.6923	0.1950	0.1057	0.2174	0.8035	0.9981	0.1490	0.3053	0.1400	0.1277
LMVSC [13]	0.5444	0.1166	0.2073	0.2485	0.6583	0.5424	0.1370	0.4420	0.3588	0.1849
2CMV [22]	0.3432	0.2371	0.1208	0.4849	0.6750	0.1001	0.1245	0.2787	0.3336	0.1865
LMSC [41]	0.5740	0.2492	0.1538	0.4091	0.7806	0.7972	0.1375	0.4820	0.3828	0.1786
OPLFMVC [21]	0.6080	0.2475	0.2198	0.4710	0.5437	0.1999	0.1405	0.2493	0.3753	0.0892
CGL [16]	0.6746	0.2683	0.1620	0.5432	0.8964	0.7211	0.1600	0.4507	0.4400	0.1942
COMVSC [44]	0.3846	0.0977	0.1062	0.2228	0.5368	0.2448	0.1215	0.2787	0.0923	0.2204
ONMVSC [17]	0.3432	0.0876	0.1974	0.2107	0.3306	0.1001	0.1530	0.2787	0.4147	0.1050
LSRMSC [10]	0.6331	-	0.1400	0.2687	0.5972	0.2605	0.1400	0.3220	0.2292	-
MvAGC [19]	0.5858	0.1461	0.1788	0.4903	0.6271	0.3122	0.1875	0.3693	0.2932	0.1218
MCLGF	0.8284	0.2758	0.2460	0.6709	0.9000	0.9966	0.1655	0.5113	0.4314	0.2616



(a) 3sources (b) Caltech (c) CCV (d) CiteSeer (e) COIL

(f) Hdigit (g) NUSWIDE (h) Reuters (i) Scene (j) SUNRGBD

Figure 2: Logarithm of the running time on all data sets (in Sec.).

comparable results. Notice that, even though LSRMSC and MvAGC both use different graph filters for the two views, they cannot handle the first view well. However, in our method, we use only *one* filter for the two views which can obtain good results for both views. This well demonstrates the superiority of the learned consensus graph filter in our method.

4.4 Ablation Study

The proposed framework involves multiple graph learning (i.e., Section 3.1) and graph filter learning (i.e., Section 3.2). In this section, we conduct the ablation study to show the effect of each part. We denote **MGL** as the degenerated version of our method which only considers multiple graph learning, i.e., the first term of Eq.(6) vanishes. We denote **GFL** as the degenerated version only considering the graph filter learning, i.e., the second term of Eq.(6) vanishes. **MCLGF** denotes the original version of our method.

Table 4 shows the ACC and NMI results of MCLGF and its two degenerated versions. On most data sets, the performance of GFL is better than MGL, which shows that graph filter learning may be more important than multiple graph learning. Moreover, MCLGF outperforms both MGL and GFL on all data sets. It demonstrates that combining graph filter learning with multiple graph learning can further improve the clustering performance.

4.5 Running Time Results

Figure 2 shows the running time of all methods on all data sets. Since on some data sets, many methods cost a lot of time, we show the logarithm of the running time, which is more readable. The rightmost black bar indicates our method. Figure 2 shows that our method is comparable with the mainstream multi-view clustering methods and even faster than some graph based multi-view clustering methods, such as CGD, COMVSC, and LSRMSC.

Table 3: NMI results on all the data sets. Red texts indicate the best results, blue texts indicate the second best results, and green texts indicate the third best results.

Methods	3sources	Caltech	CCV	CiteSeer	COIL	Hdigit	NUSWIDE	Reuters	Scene	SUNRGBD
RMSC [36]	0.4177	0.2037	0.2013	0.0156	0.6838	0.3108	0.1941	0.0699	0.1000	0.0841
AMGL [25]	0.0583	0.3079	0.1611	0.0031	0.9299	0.9250	0.1728	0.0265	0.3247	0.1758
MVGL [40]	0.0660	0.1609	0.0159	0.0177	0.9104	0.9866	0.0688	0.0602	0.1584	0.0368
AWP [26]	0.3790	0.4526	0.1202	0.0489	0.8473	0.7706	0.1650	0.0619	0.3564	0.2074
MCGC [39]	0.0607	0.1867	0.0035	0.0014	0.8997	0.0009	0.0481	0.0634	0.0912	0.0862
CGD [29]	0.6939	0.4435	0.1173	0.1059	0.8636	0.7241	0.1550	0.2878	0.4147	0.2409
GMC [34]	0.5480	0.2379	0.0022	0.0072	0.9176	0.9939	0.0852	0.0883	0.0582	0.0402
LMVSC [13]	0.3614	0.2573	0.1681	0.0464	0.7804	0.4999	0.1410	0.2898	0.3493	0.2329
2CMV [22]	0.0308	0.4355	0.0493	0.2519	0.7845	0.0009	0.0156	0.0040	0.3189	0.2481
LMSC [41]	0.4775	0.4608	0.1129	0.2195	0.8421	0.7958	0.1527	0.3444	0.3500	0.2274
OPLFMVC [21]	0.5290	0.4239	0.1615	0.2260	0.7131	0.1087	0.1491	0.0039	0.3815	0.0418
CGL [16]	0.6780	0.4986	0.1174	0.2724	0.9364	0.8394	0.1809	0.2595	0.4115	0.2621
COMVSC [44]	0.1179	0.0362	0.0030	0.0107	0.7064	0.1963	0.0151	0.0069	0.0032	0.1218
ONMVSC [17]	0.0443	0.0082	0.1699	0.0039	0.4690	0.0009	0.1919	0.0040	0.4013	0.0034
LSRMSC [10]	0.4522	-	0.0727	0.0302	0.7099	0.1600	0.1313	0.0476	0.1786	-
MvAGC [19]	0.5511	0.2937	0.1094	0.2640	0.7145	0.1774	0.1961	0.0594	0.2616	0.1369
MCLGF	0.6990	0.4570	0.2017	0.4078	0.9441	0.9897	0.2001	0.2860	0.4192	0.2796

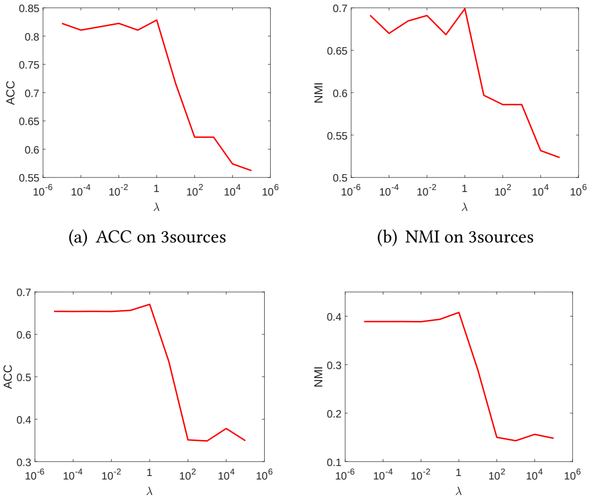
Table 4: Ablation Study.

Data sets	MGL		GFL		MCLGF	
	ACC	NMI	ACC	NMI	ACC	NMI
3sources	0.5799	0.5743	0.6331	0.6030	**0.8284**	**0.6990**
Caltech	0.2605	0.4285	0.2306	0.4509	**0.2758**	**0.4570**
CCV	0.1887	0.1691	0.2291	0.1834	**0.2460**	**0.2017**
CiteSeer	0.2681	0.0816	0.6543	0.3890	**0.6709**	**0.4078**
COIL	0.8715	0.9344	0.7333	0.8514	**0.9000**	**0.9441**
Hdigit	0.8529	0.9158	0.9865	0.9793	**0.9966**	**0.9897**
NUSWIDE	0.1355	0.1648	0.1650	0.1943	**0.1655**	**0.2001**
Reuters	0.3920	0.1279	0.4540	0.2488	**0.5113**	**0.2860**
Scene	0.3373	0.3510	0.4297	0.4099	**0.4314**	**0.4192**
SUNRGBD	0.1846	0.2334	0.1933	0.2400	**0.2616**	**0.2796**

Despite this, in the future, we will study how to further speed up it to handle larger data.

4.6 Parameter Study

In this subsection, we show the effect of the hyper-parameter λ. We tune λ in $[10^{-5}, 10^{5}]$. Figure 3 show the ACC and NMI results on 3sources and CiteSeer data sets. The results on other data sets are similar. Figure 3 shows that the proposed MCLGF can achieve relatively good results when $\lambda \leq 1$. Notice that λ is a hyperparameter to control the weights of multiple graph learning and graph filter learning. When it is small, which means the graph filter learning will have a larger weight than multiple graph learning, the method can achieve better performance. It means that the graph filter learning is more important than multiple graph learning which is consistent with the results of the ablation study.



(a) ACC on 3sources

(b) NMI on 3sources

(c) ACC on CiteSeer

(d) NMI on CiteSeer

Figure 3: Clustering results of w.r.t. λ on 3sources and CiteSeer data sets.

5 CONCLUSION

This paper proposes a novel multi-view clustering method with a learnable graph filter. Different from other graph filter based multi-view clustering methods, which directly use pre-defined graph filters, our method focuses on how to leverage the information in all views to learn an appropriate consensus graph filter for clustering. To this end, we design a framework of graph filter learning with multiple graph learning. We also provide an iterative algorithm to learn the graph filter and do the multi-view clustering. At last, we conduct extensive experiments by comparing with some state-of-the-art multi-view clustering methods to demonstrate the effectiveness and superiority of the proposed method.

ACKNOWLEDGMENTS

This work is supported by the National Natural Science Foundation of China grants 62176001, 61806003, and 61976129.

REFERENCES

[1] Chidanand Apté, Fred Damerau, and Sholom M. Weiss. 1994. Automated Learning of Decision Rules for Text Categorization. *ACM Trans. Inf. Syst.* 12, 3 (1994), 233–251.

[2] Stephen Boyd, Neal Parikh, Eric Chu, Borja Peleato, Jonathan Eckstein, et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning* 3, 1 (2011), 1–122.

[3] Xiao Cai, Feiping Nie, and Heng Huang. 2013. Multi-View K-Means Clustering on Big Data. In *IJCAI 2013, Proceedings of the 23rd International Joint Conference on Artificial Intelligence, Beijing, China, August 3-9, 2013*, Francesca Rossi (Ed.). IJCAI/AAAI, 2598–2604.

[4] Tat-Seng Chua, Jinhui Tang, Richang Hong, Haojie Li, Zhiping Luo, and Yan-Tao Zheng. July 8-10, 2009. NUS-WIDE: A Real-World Web Image Database from National University of Singapore. In *Proc. of ACM Conf. on Image and Video Retrieval (CIVR'09)*. Santorini, Greece.

[5] Laurent Condat. 2016. Fast projection onto the simplex and the l_1 ball. *Math. Program.* 158, 1-2 (2016), 575–585. https://doi.org/10.1007/s10107-015-0946-6

[6] Xiaowen Dong, Dorina Thanou, Pascal Frossard, and Pierre Vandergheynst. 2016. Learning Laplacian Matrix in Smooth Graph Signal Representations. *IEEE Transactions on Signal Processing* 64, 23 (2016), 6160–6173. https://doi.org/10.1109/TSP.2016.2602809

[7] Jing Gao, Jiawei Han, Jialu Liu, and Chi Wang. 2013. Multi-View Clustering via Joint Nonnegative Matrix Factorization. In *Proceedings of the 13th SIAM International Conference on Data Mining, May 2-4, 2013. Austin, Texas, USA.* SIAM, 252–260.

[8] C. Lee Giles, Kurt D. Bollacker, and Steve Lawrence. 1998. CiteSeer: An Automatic Citation Indexing System. In *Proceedings of the Third ACM Conference on Digital Libraries* (Pittsburgh, Pennsylvania, USA) *(DL '98)*. Association for Computing Machinery, New York, NY, USA, 89–98.

[9] Zongbo Han, Changqing Zhang, Huazhu Fu, and Joey Tianyi Zhou. 2023. Trusted Multi-View Classification With Dynamic Evidential Fusion. *IEEE Trans. Pattern Anal. Mach. Intell.* 45, 2 (2023), 2551–2566.

[10] Shudong Huang, Yixi Liu, Yazhou Ren, Ivor W. Tsang, Zenglin Xu, and Jiancheng Lv. 2022. Learning Smooth Representation for Multi-view Subspace Clustering. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022.* ACM, 3421–3429.

[11] Zongmo Huang, Yazhou Ren, Xiaorong Pu, and Lifang He. 2021. Non-Linear Fusion for Self-Paced Multi-View Clustering. In *MM '21: ACM Multimedia Conference, Virtual Event, China, October 20 - 24, 2021*, Heng Tao Shen, Yueting Zhuang, John R. Smith, Yang Yang, Pablo César, Florian Metze, and Balakrishnan Prabhakaran (Eds.). ACM, 3211–3219.

[12] Zhao Kang, Zhanyu Liu, Shirui Pan, and Ling Tian. 2022. Fine-grained Attributed Graph Clustering. In *Proceedings of the 2022 SIAM International Conference on Data Mining, SDM 2022, Alexandria, VA, USA, April 28-30, 2022.* SIAM, 370–378.

[13] Zhao Kang, Wangtao Zhou, Zhitong Zhao, Junming Shao, Meng Han, and Zenglin Xu. 2020. Large-Scale Multi-View Subspace Clustering in Linear Time. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020.* AAAI Press, 4412–4419.

[14] Johannes Klicpera, Stefan Weißenberger, and Stephan Günnemann. 2019. Diffusion Improves Graph Learning. In *Advances in Neural Information Processing Systems 32: Annual Conference on Neural Information Processing Systems 2019, NeurIPS 2019, December 8-14, 2019, Vancouver, BC, Canada.* 13333–13345.

[15] Abhishek Kumar, Piyush Rai, and Hal Daumé III. 2011. Co-regularized Multi-view Spectral Clustering. In *Advances in Neural Information Processing Systems 24: 25th Annual Conference on Neural Information Processing Systems 2011. Proceedings of a meeting held 12-14 December 2011, Granada, Spain*, John Shawe-Taylor, Richard S. Zemel, Peter L. Bartlett, Fernando C. N. Pereira, and Kilian Q. Weinberger (Eds.). 1413–1421.

[16] Zhenglai Li, Chang Tang, Xinwang Liu, Xiao Zheng, Wei Zhang, and En Zhu. 2022. Consensus Graph Learning for Multi-View Clustering. *IEEE Trans. Multim.* 24 (2022), 2461–2472.

[17] Weixuan Liang, Sihang Zhou, Jian Xiong, Xinwang Liu, Siwei Wang, En Zhu, Zhiping Cai, and Xin Xu. 2022. Multi-View Spectral Clustering With High-Order Optimal Neighborhood Laplacian Matrix. *IEEE Trans. Knowl. Data Eng.* 34, 7 (2022), 3418–3430.

[18] Zhiping Lin and Zhao Kang. 2021. Graph Filter-based Multi-view Attributed Graph Clustering. In *Proceedings of the Thirtieth International Joint Conference on Artificial Intelligence, IJCAI 2021, Virtual Event / Montreal, Canada, 19-27 August 2021.* ijcai.org, 2723–2729.

[19] Zhiping Lin, Zhao Kang, Lizong Zhang, and Ling Tian. 2023. Multi-View Attributed Graph Clustering. *IEEE Trans. Knowl. Data Eng.* 35, 2 (2023), 1872–1880.

[20] Dong C. Liu and Jorge Nocedal. 1989. On the limited memory BFGS method for large scale optimization. *Math. Program.* 45, 1-3 (1989), 503–528.

[21] Xinwang Liu, Li Liu, Qing Liao, Siwei Wang, Yi Zhang, Wenxuan Tu, Chang Tang, Jiyuan Liu, and En Zhu. 2021. One Pass Late Fusion Multi-view Clustering. In *Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event (Proceedings of Machine Learning Research, Vol. 139).* PMLR, 6850–6859.

[22] Khanh Luong and Richi Nayak. 2020. A Novel Approach to Learning Consensus and Complementary Information for Multi-View Data Clustering. In *36th IEEE International Conference on Data Engineering, ICDE 2020, Dallas, TX, USA, April 20-24, 2020.* IEEE, 865–876.

[23] Zhengrui Ma, Zhao Kang, Guangchun Luo, Ling Tian, and Wenyu Chen. 2020. Towards Clustering-friendly Representations: Subspace Clustering via Graph Filtering. In *MM '20: The 28th ACM International Conference on Multimedia, Virtual Event / Seattle, WA, USA, October 12-16, 2020.* ACM, 3081–3089.

[24] Feiping Nie, Guohao Cai, Jing Li, and Xuelong Li. 2018. Auto-Weighted Multi-View Learning for Image Clustering and Semi-Supervised Classification. *IEEE Trans. Image Process.* 27, 3 (2018), 1501–1511.

[25] Feiping Nie, Jing Li, and Xuelong Li. 2016. Parameter-Free Auto-Weighted Multiple Graph Learning: A Framework for Multiview Clustering and Semi-Supervised Classification. In *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence, IJCAI 2016, New York, NY, USA, 9-15 July 2016*, Subbarao Kambhampati (Ed.). IJCAI/AAAI Press, 1881–1887.

[26] Feiping Nie, Lai Tian, and Xuelong Li. 2018. Multiview Clustering via Adaptively Weighted Procrustes. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, KDD 2018, London, UK, August 19-23, 2018*, Yike Guo and Faisal Farooq (Eds.). ACM, 2022–2030.

[27] Erlin Pan and Zhao Kang. 2021. Multi-view Contrastive Graph Clustering. In *NeurIPS.*

[28] David I. Shuman, Sunil K. Narang, Pascal Frossard, Antonio Ortega, and Pierre Vandergheynst. 2013. The Emerging Field of Signal Processing on Graphs: Extending High-Dimensional Data Analysis to Networks and Other Irregular Domains. *IEEE Signal Process. Mag.* 30, 3 (2013), 83–98.

[29] Chang Tang, Xinwang Liu, Xinzhong Zhu, En Zhu, Zhigang Luo, Lizhe Wang, and Wen Gao. 2020. CGD: Multi-View Clustering via Cross-View Graph Diffusion. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020.* AAAI Press, 5924–5931.

[30] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. 2017. From Ensemble Clustering to Multi-View Clustering. In *IJCAI.* 2843–2849.

[31] Zhiqiang Tao, Hongfu Liu, Sheng Li, Zhengming Ding, and Yun Fu. 2019. Robust Spectral Ensemble Clustering via Rank Minimization. *ACM Transactions on Knowledge Discovery From Data* 13, 1 (2019), 1–25.

[32] Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing Data using t-SNE. *Journal of Machine Learning Research* 9, 86 (2008), 2579–2605.

[33] Hoi-To Wai, Santiago Segarra, Asuman E. Ozdaglar, Anna Scaglione, and Ali Jadbabaie. 2020. Blind Community Detection From Low-Rank Excitations of a Graph Filter. *IEEE Trans. Signal Process.* 68 (2020), 436–451.

[34] Hao Wang, Yan Yang, and Bing Liu. 2020. GMC: Graph-Based Multi-View Clustering. *IEEE Trans. Knowl. Data Eng.* 32, 6 (2020), 1116–1129.

[35] Jie Wen, Ke Yan, Zheng Zhang, Yong Xu, Junqian Wang, Lunke Fei, and Bob Zhang. 2021. Adaptive Graph Completion Based Incomplete Multi-View Clustering. *IEEE Trans. Multim.* 23 (2021), 2493–2504.

[36] Rongkai Xia, Yan Pan, Lei Du, and Jian Yin. 2014. Robust Multi-View Spectral Clustering via Low-Rank and Sparse Decomposition. In *Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, July 27 -31, 2014, Québec City, Québec, Canada.* AAAI Press, 2149–2155.

[37] Jie Xu, Yazhou Ren, Huayi Tang, Zhimeng Yang, Lili Pan, Yang Yang, Xiaorong Pu, Philip S. Yu, and Lifang He. 2022. Self-Supervised Discriminative Feature Learning for Deep Multi-View Clustering. *IEEE Transactions on Knowledge and Data Engineering* (2022), 1–12.

[38] Jie Xu, Huayi Tang, Yazhou Ren, Liang Peng, Xiaofeng Zhu, and Lifang He. 2022. Multi-level Feature Learning for Contrastive Multi-view Clustering. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022.* IEEE, 16030–16039.

[39] Kun Zhan, Feiping Nie, Jing Wang, and Yi Yang. 2019. Multiview Consensus Graph Clustering. *IEEE Trans. Image Process.* 28, 3 (2019), 1261–1270.

[40] Kun Zhan, Changqing Zhang, Junpeng Guan, and Junsheng Wang. 2018. Graph Learning for Multiview Clustering. *IEEE Trans. Cybern.* 48, 10 (2018), 2887–2895.

[41] Changqing Zhang, Huazhu Fu, Qinghua Hu, Xiaochun Cao, Yuan Xie, Dacheng Tao, and Dong Xu. 2020. Generalized Latent Multi-View Subspace Clustering. *IEEE Trans. Pattern Anal. Mach. Intell.* 42, 1 (2020), 86–99.

[42] Changqing Zhang, Qinghua Hu, Huazhu Fu, Pengfei Zhu, and Xiaochun Cao. 2017. Latent Multi-view Subspace Clustering. In *2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017.* IEEE Computer Society, 4333–4341.

[43] Guang-Yu Zhang, Dong Huang, and Chang-Dong Wang. 2023. Facilitated low-rank multi-view subspace clustering. *Knowl. Based Syst.* 260 (2023), 110141.

[44] Pei Zhang, Xinwang Liu, Jian Xiong, Sihang Zhou, Wentao Zhao, En Zhu, and Zhiping Cai. 2022. Consensus One-Step Multi-View Subspace Clustering. *IEEE Trans. Knowl. Data Eng.* 34, 10 (2022), 4676–4689.

[45] Nan Zhao and Jie Bu. 2022. Robust multi-view subspace clustering based on consensus representation and orthogonal diversity. *Neural Networks* 150 (2022), 102–111.

[46] Bolei Zhou, Àgata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. 2014. Learning Deep Features for Scene Recognition using Places Database. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, Zoubin Ghahramani, Max Welling, Corinna Cortes, Neil D. Lawrence, and Kilian Q. Weinberger (Eds.). 487–495.

[47] Dengyong Zhou and Bernhard Schölkopf. 2004. A Regularization Framework for Learning from Graph Data. In *Proc. ICML Workshop Statist. Relational Learn.* 132–137.

[48] Peng Zhou, Liang Du, and Xuejun Li. 2020. Self-paced Consensus Clustering with Bipartite Graph. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence, IJCAI 2020*. 2133–2139.

[49] Peng Zhou, Liang Du, Yi-Dong Shen, and Xuejun Li. 2021. Tri-level Robust Clustering Ensemble with Multiple Graph Learning. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021*. 11125–11133.

[50] Peng Zhou, Xinwang Liu, Liang Du, and Xuejun Li. 2023. Self-paced Adaptive Bipartite Graph Learning for Consensus Clustering. *ACM Trans. Knowl. Discov. Data* 17, 5 (2023), 62:1–62:35.

[51] Peng Zhou, Yi-Dong Shen, Liang Du, and Fan Ye. 2019. Incremental Multi-view Support Vector Machine. In *Proceedings of the 2019 SIAM International Conference on Data Mining, SDM 2019, Calgary, Alberta, Canada, May 2-4, 2019*, Tanya Y. Berger-Wolf and Nitesh V. Chawla (Eds.). SIAM, 1–9.

[52] Peng Zhou, Yi-Dong Shen, Liang Du, Fan Ye, and Xuejun Li. 2019. Incremental multi-view spectral clustering. *Knowl. Based Syst.* 174 (2019), 73–86.

[53] Linlin Zong, Xianchao Zhang, Xinyue Liu, and Hong Yu. 2018. Weighted Multi-View Spectral Clustering Based on Spectral Perturbation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*. 4621–4629.