Neural Networks 144 (2021) 260–270

Contents lists available at ScienceDirect

Neural Networks

journal homepage: www.elsevier.com/locate/neunet

Incremental multi-view spectral clustering with sparse and connected graph learning

Hongwei Yin [a,b,*], Wenjun Hu [a,b,**], Zhao Zhang [c], Jungang Lou [a,b], Minmin Miao [a,b]

[a] School of Information Engineering, Huzhou University, Hu'zhou 313000, China
[b] Zhejiang Province Key Laboratory of Smart Management & Application of Modern Agricultural Resources, Huzhou University, Hu'zhou 313000, China
[c] School of Computer Science and Information Engineering & Key Laboratory of Knowledge Engineering with Big Data (Ministry of Education), Hefei University of Technology, He'fei 230009, China

ARTICLE INFO

Article history:
Received 4 January 2021
Received in revised form 1 June 2021
Accepted 26 August 2021
Available online 5 September 2021

Keywords:
Multi-view clustering
Incremental clustering
Sparse graph learning
Connected graph learning
Spectral embedding

ABSTRACT

In recent years, a lot of excellent multi-view clustering methods have been proposed. Because most of them need to fuse all views at one time, they are infeasible as the number of views increases over time. If the present multi-view clustering methods are employed directly to re-fuse all views at each time, it is too expensive to store all historical views. In this paper, we proposed an efficient incremental multi-view spectral clustering method with sparse and connected graph learning (SCGL). In our method, only one consensus similarity matrix is stored to represent the structural information of all historical views. Once the newly collected view is available, the consensus similarity matrix is reconstructed by learning from its previous version and the current new view. To further improve the incremental multi-view clustering performance, the sparse graph learning and the connected graph learning are integrated into our model, which can not only reduce the noises, but also preserve the correct connections within clusters. Experiments on several multi-view datasets demonstrate that our method is superior to traditional methods in clustering accuracy, and is more suitable to deal with the multi-view clustering with the number of views increasing over time.

1. Introduction

With the rapid development of information technology, data can often be obtained from multiple sources. Compared with the traditional single-source data, the information contained in multi-source data is much richer. Many previous studies have shown that combining information from multiple sources can effectively improve the ability of machine learning algorithms (Fu, Lin, Vasilakos, & Wang, 2020). Clustering is a fundamental task in machine learning, which aims to partition input data into different groups. In recent years, many researches focus on how to explore and exploit the complementary information from multiple sources to produce a more robust and precise partitioning of the data than traditional clustering methods. This type of work is called multi-view clustering (Wen, Zhang, Zhang, Zhu et al., 2020).

According to the different data representation models, the main work of multi-view clustering can be roughly divided into four categories, including multi-view subspace clustering (Wang, Lei, Guo, & Zhang, 2019; Wang, Wang, Zhang, Fu, & Wag, 2020), multi-view nonnegative matrix factorization (Liu, Wang, Gao, & Han, 2013; Zong, Zhang, Zhao, Yu, & Zhao, 2017), deep multi-view clustering (Li, Wang, Tao, Gao, & Yang, 2019; Wen, Zhang, Zhang, Wu et al., 2020), and graph-based multi-view clustering (Gao et al., 2020; Huang, Xu, Tsang, & Kang, 2020). Compared with other methods, graph-based multi-view clustering methods have more advantages in theoretical support and clustering precision. At first, an unified graph is typically obtained by fusing the input graphs of all views, where the structure information of multi-view data is represented by the graph model. Then, an additional clustering algorithm is employed on this unified graph to produce the final clustering results, such as spectral clustering (von Luxburg, 2007). A lot of related works have achieved excellent performances for multi-view clustering task, which proves the effectiveness of this learning paradigm (Hu, Nie, Chang et al., 2020; Wen, Zhang, Zhang, Fei, & Wang, 2021; Zhan, Zhang, Guan, & Wang, 2018).

However, in many practical applications, the number of views is not a fixed value, and it increases over time. For example, in brain computer interfaces system, brain signal acquisition plays an important role for analyzing brain states. Since the brain

* School of Information Engineering, Huzhou University, Hu'zhou 313000, China.
** Corresponding author.
E-mail addresses: 02713@zjhu.edu.cn (H. Yin), hoowenjun@foxmail.com (W. Hu).

https://doi.org/10.1016/j.neunet.2021.08.031
0893-6080/

signal changes with the subject's mental state, it is necessary to collect the signal at different times. Therefore, the signal data at each time constitutes a view, and the number of views increases over time (Miao, Zhang, Hu, & Wang, 2020). The present multi-view clustering methods have two problems when the number of views increases over time. First, it is difficult to guarantee that all views are available at one time. The traditional methods need to fuse all views at one time, which means they are infeasible for the newly collected views. Second, if we directly employ the present multi-view clustering methods to re-fuse all views at each time, it is too expensive to store all historical views.

In order to solve such problems with the number of views increasing over time, this paper proposes an effective incremental multi-view clustering method with sparse and connected graph learning (SCGL). In our method, one unified graph is learned to represent the structure of previous views. When a newly collected view is available, our task is to update this unified graph by learning from its previous version and the newly collected view. The main contributions of this paper are summarized as follows:

1. This paper proposes a novel method to complete multi-view clustering task with the number of views increasing. Considering the newly collected view as increment, our method can fuse different views one by one instead of fusing them at one time. Therefore, our method is able to dynamically fuse newly collected views. We utilize only one unified graph to represent the structure information of all previous views, which greatly reduces the space cost and enhances the scalability of the method.
2. In order to improve the accuracy of incremental multi-view spectral clustering, the sparse and connected graph learning are incorporated in our model to maintain a clear diagonal block structure of the graph. First, We introduce a l_1 norm regularizer to satisfy the unified graph sparsity. Then, the connections between points in spectral embedding are utilized to reconstruct the unified graph.
3. To verify the effectiveness of our method, we conduct experiments on several benchmark datasets. Compared with the state-of-the-art methods, our method performs better for incremental multi-view clustering.

The main notations are summarized in Table 1. In this paper, matrices are represented in bold upper case, vectors are represented in bold lower case and scalars are represented in lower case. For a matrix $\boldsymbol{W}$, w_{ij} is the element in the ith row and the jth column. The trace of $\boldsymbol{W}$ is denoted by $Tr(\boldsymbol{W})$. The Frobenius norm of $\boldsymbol{W}$ is denoted by $\|\boldsymbol{W}\|_F$ and the l_1 norm of $\boldsymbol{W}$ is denoted by $\|\boldsymbol{W}\|_1$. The identity matrix is denoted by $\boldsymbol{I}$. The rest of this paper is organized as follows.

The related works are introduced in Section 2. The details of our method are described in Section 3. Experiment results on several benchmark datasets are reported in Section 4. Finally, Section 5 concludes the paper.

Table 1
Main notation.

n	Number of samples
d	Dimension of features
m	Number of views
c	Number of clusters
$\boldsymbol{X} = \{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_n\}$	Data matrix
$\boldsymbol{X}^{(v)} = \{\boldsymbol{x}_1^{(v)}, \ldots, \boldsymbol{x}_n^{(v)}\}$	Data matrix of vth view
$\boldsymbol{W}^{(v)}$	Similarity matrix of vth view
$\widetilde{\boldsymbol{W}}_t$	consensus similarity matrix at time t
$\|\boldsymbol{W}\|_F = \sqrt{\sum_{i,j} \boldsymbol{w}_{i,j}^2}$	Frobenius norm of $\boldsymbol{W}$
$\|\boldsymbol{W}\|_1 = \sum_{i,j} \|\boldsymbol{w}_{i,j}\|$	l_1 norm of $\boldsymbol{W}$
$\boldsymbol{I}$	Identity matrix

2. Related work

In this section, we briefly review some representative multi-view clustering methods and the single view spectral clustering that are related to our method.

2.1. Multi-view clustering

For the past few years, numerous multi-view clustering methods have been proposed. Although such methods follow the same way by using the complementary information of different views to improve the clustering accuracy, there is still no definite paradigm for the data representation and the corresponding fusion. Generally, existing methods can be roughly divided into four categories, including multi-view subspace clustering, multi-view nonnegative matrix factorization, deep multi-view clustering, and graph-based multi-view clustering.

Gao, Nie, Li, and Huang (2015) first extend subspace clustering to multi-view clustering by performing subspace clustering on each view simultaneously. They employ a common cluster indicator to keep the consistence between different views. Zhu, Lu, and Zhou (2019) propose a structured multi-view subspace clustering (SMSC) to explore the structural shared and general properties of different views. SMSC employs a structure consistence constraint on the general subspace to preserve the similarity information of each view and utilizes a novel diversity regularizer to enhance the diversity of different subspaces. Zhang, Liu, Shen, Shen, and Shao (2019) propose a binary multi-view clustering method (BMVC) on large-scale dataset by uniting the collaborative binary codes of different views and binary cluster structures into a joint framework.

In multi-view nonnegative matrix factorization, Liu et al. (2013) first propose to integrate the coefficient matrices of different views and obtain a consensus nonnegative embedding. Because the nonnegative matrix factorization ignores the local structure of original data, Zhang, Zhao, Zong, Liu, and Yu (2014) use a multi-manifold regularization to preserve the locally manifold structure of multi-view data. Gao, Yu, Jin, and Yin (2019) integrate the multi-manifold regularization and low-rank constraint into a framework. The common clusters of different views are found by the low-rank nonnegative matrix factorization. To learn high-level complementary information of multi-view data, Li et al. (2020) construct a deep NMF model. The graph regularization is incorporated into all layers to improve the representation ability. Li, Wang et al. (2019) propose a deep adversarial multi-view clustering network, which adopts deep auto-encoders and adversarial training to learn the latent non-linear structure of multi-view data.

In graph-based methods, the main strategy is to extend the classical single view spectral clustering to multi-view scenarios. Kumar and III (2011) propose a co-training approach to make the spectral embeddings of different views consistent. In addition, They propose a co-regularized approach by minimizing the disagreement between the spectral embeddings of different views (Kumar, Rai, & III, 2011). Yin, Li, Zhang, and Zhang (2019) propose a multi-view spectral clustering via spectral embedding fusion by integrating the global structure information and local structure information. Li, Tang et al. (2019) constrain both consistency and diversity in label distribution to improve the performance of multi-view clustering. Hu, Nie, Chang et al. (2020) propose an effective method for multi-view clustering with sparse graph learning (SMVSC). The close form solution can be obtained directly without iteration in their model. For these methods, the final result is generally returned by conducting

additional clustering to the spectral embedding, which inevitably leads to a suboptimum solution. Hu, Nie, Wang, and Li (2020) propose a method to avoid the post-processing by learning a consistent nonnegative embedding and spectral embedding simultaneously, where the clustering result is directly revealed by the nonnegative embedding.

2.2. Incremental clustering

In real world applications, information is hard to gather all at once. Instead, the information is collected incrementally over time, where it creates the need for incremental learning. Under the scenario of incremental clustering, data stream becomes available over time. Accordingly, it requires to utilize the historical data over time to benefit future clustering (He, Chen, Li, & Xu, 2011). Mehrkanoon, Agudelo, and Suykens (2015) make kernel spectral clustering algorithm applicable for data stream, where Kalman filter is integrated as a regularizer by providing an estimation of the labels throughout the whole data stream. Liu and Ban (2015) propose a novel method called growing incremental self-organizing neural network. In this method, a two-dimensional topological graph is used to represent the information about both cluster categories and the detailed relationship within cluster. In Roy, Panda, and Roy (2020), a network made of CNNs is proposed that grows and learns as new data becomes available. In this network, the new classes are added to the hierarchical structure like new leaves.

Although the current incremental learning methods are relatively rich, few of them are able to cope with the newly collected views. Zhou, Shen, Du, Ye, and Li (2019) propose an incremental multi-view clustering method by integrating different views one by one. In this method, a set of base kernels is preserved to represent the collection of previous views. When one new view is available, they need to update all base kernels simultaneously with iterative optimization method. This approach is space-consuming, with too many parameters leading to complex implementation. At the same time, there is a major challenge for incremental clustering called stability-plasticity dilemma. On the one hand, the clustering results should be plastic to new input data from non-stationary distributions. On the other hand, the clustering results should retain performance of previous input data (Rosenfeld & Tsotsos, 2020).

In this paper, we propose a simple yet efficient incremental multi-view spectral clustering in which only one unified graph is preserved to represent all the previous views. When a new view is arriving, our method updates the unified graph by learning from the previous version and the new view. Meanwhile, the contribution of a view depends on its clustering ability, which determines its corresponding weight. This weight setting is a way out of the stability-plasticity dilemma. Besides, a clear diagonal block structure of the unified graph is maintained by sparse and connected graph learning, which can improve the accuracy of clustering.

2.3. Spectral clustering

Given a set of data points $\boldsymbol{X} = \{\boldsymbol{x}_1, \boldsymbol{x}_2, \ldots, \boldsymbol{x}_n\} \in \mathcal{R}^{d\times n}$, the goal of clustering is to find an optimal partition of $\boldsymbol{X}$, where the points in same cluster have larger similarities and those in different clusters have smaller similarities (Wang, Zhu et al., 2020). The clustering result is denoted by a discrete indicator matrix $\boldsymbol{Y} \in \mathcal{R}^{n\times c}$, where c is the number of clusters. Each point is treated as a node on graph in spectral clustering. Then, the dataset is transformed into an undirected weighted graph $\mathcal{G} = \{\mathcal{V}, \mathcal{E}\}$, where $\mathcal{V}$ is the node set and $\mathcal{E}$ is the edge set. The similarity between the ith node and the jth node is denoted by $w_{i,j}$, and the similarity between nodes is represented by a similarity matrix $\boldsymbol{W} \in \mathcal{R}^{n\times n}$. The similarity matrix is defined as follows:

$$w_{ij} = \begin{cases} exp(-\frac{\|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2}{2\sigma^2}), & \text{if } \boldsymbol{x}_i \in \mathcal{N}(\boldsymbol{x}_j) \text{ or } \boldsymbol{x}_j \in \mathcal{N}(\boldsymbol{x}_i) \\ 0, & \text{otherwise} \end{cases} \tag{1}$$

where $\mathcal{N}(\cdot)$ is the function for searching K-nearest neigh- bors and σ is a bandwidth parameter that controls the spread of the neighbors. Let $\boldsymbol{D} = diag\ (d_1, d_2, \ldots, d_n)$ denote the degree matrix, where $d_i = \sum_{j=1}^{n} w_{ij}$ is the degree of ith node. Spectral clustering transforms the clustering problem into a normalized cut problem on the graph, and the objective function of spectral clustering is defined as (von Luxburg, 2007):

$$\begin{aligned} &\max_{\boldsymbol{F}}\ tr\left(\boldsymbol{F}^T\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}\boldsymbol{F}\right) \\ &s.t.\ \ \boldsymbol{F}^T\boldsymbol{F} = \boldsymbol{I} \end{aligned} \tag{2}$$

here the matrix $\boldsymbol{F}$ is often referred to as spectral embedding, which consists of the eigenvectors corresponding to the top k eigenvalues of matrix $\boldsymbol{D}^{-1/2}\boldsymbol{W}\boldsymbol{D}^{-1/2}$. Because the spectral embedding is a relaxed solution of the discrete label, the final clustering result is returned by conducting K-means on it.

3. Proposed method

In this section, our goal is to fuse different views one by one in an incremental strategy. First, we provide an incremental multi-view sparse graph learning model. Secondly, we use the connected graph learning to maintain the correct connections within each cluster. The basic framework is shown in Fig. 1.

3.1. Incremental multi-view spectral clustering with sparse graph learning

As discussed in spectral clustering, the structure information of graph is represented by the similarity matrix. In multi-view spectral clustering, a consensus similarity matrix is learned by fusing the similarity matrices of different views. The consensus similarity matrix can represent the structure information hidden in the multi-view data. Let $\{\boldsymbol{X}^{(v)}\}_{v=1}^{m}$ denote the multi-view data with m views. The similarity matrices of different views are constructed by gaussian kernel function separately. The learning model of the consensus similarity matrix is defined as follows (Hu, Nie, Chang et al., 2020):

$$\min_{\widetilde{\boldsymbol{W}}} \sum_{v=1}^{m} \|\widetilde{\boldsymbol{W}} - \boldsymbol{W}^{(v)}\|_F^2 + \lambda\|\widetilde{\boldsymbol{W}}\|_1 \tag{3}$$

where $\widetilde{\boldsymbol{W}}$ denotes the consensus similarity matrix and $\boldsymbol{W}^{(v)}$ denotes the similarity matrix of vth view. In order to alleviate the noise generated by different views, a l_1 regularization is adopted as the sparse constraint.

Incremental multi-view spectral clustering states that all views should be fused one by one. Once the new view is available, we aim to make the new consensus similarity matrix as much as consistent with the previous version and the similarity matrix of the new view. Based on the model in Eq. (3), the incremental learning model of the consensus similarity matrix at time $t+1$ is defined as follows:

$$\min_{\widetilde{\boldsymbol{W}}_{t+1}} \|\widetilde{\boldsymbol{W}}_{t+1} - \widetilde{\boldsymbol{W}}_t\|_F^2 + \|\widetilde{\boldsymbol{W}}_{t+1} - \boldsymbol{W}^{(t+1)}\|_F^2 + \lambda\|\widetilde{\boldsymbol{W}}_{t+1}\|_1 \tag{4}$$

where $\widetilde{\boldsymbol{W}}_{t+1}$ is the new consensus similarity matrix at time t, $\widetilde{\boldsymbol{W}}_t$ is the previous consensus similarity matrix, and $\boldsymbol{W}^{(t+1)}$ is the similarity matrix of the new $(t+1)$-th view. The first term is to make the new consensus similarity matrix as much as consistent with the previous consensus similarity matrix. The second term is to make the new consensus similarity matrix as much as consistent with the similarity matrix of the current new view. The third term is a sparse regularization by using l_1 norm.

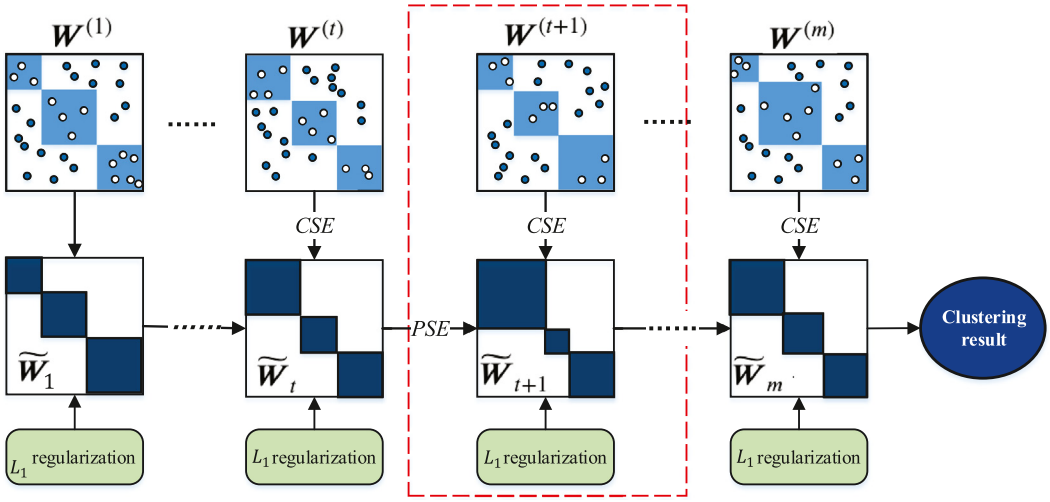


Fig. 1. The basic framework of our proposed incremental multi-view spectral clustering method. Here PSE is the spectral embedding of the previous consensus similarity matrix, CSE is the spectral embedding of the current similarity matrix. Once the $(t+1)$-th view is available, the new consensus similarity matrix $\widetilde{\boldsymbol{W}}_{t+1}$ is learned by using the PSE of $\widetilde{\boldsymbol{W}}_t$, the CSE of $\boldsymbol{W}^{(t+1)}$ and the L_1 regularization. Our method fuses the views one by one until no more new view is available.

3.2. Incremental multi-view spectral clustering with sparse and connected graph learning

In the above model, the sparse regularization can alleviate the noise generated by different views. In other words, the sparse regularization can reduce the false connections between clusters. However, excessively sparsity will sacrifice correct connections within each clusters and thus reduce the clustering performance. Therefore, we aim to preserve the correct connections within each clusters, which is called connected graph learning.

Generally, the spectral embedding of similarity matrix can be written as $\boldsymbol{F} = \{\boldsymbol{f}_1, \boldsymbol{f}_2, \ldots, \boldsymbol{f}_n\}^T$. Each row of the spectral embedding represents a point of original data e.g., the ith point and jth point is represented as $\boldsymbol{f}_i$ and $\boldsymbol{f}_j$ respectively. When the ith point and the jth point belong to the same cluster, $\boldsymbol{f}_i$ should be similar to $\boldsymbol{f}_j$ and $\boldsymbol{f}_i^T\boldsymbol{f}_j$ will take a large value. Conversely, when the ith point and the jth point belong to the different cluster, $\boldsymbol{f}_i$ should be dissimilar to $\boldsymbol{f}_j$ and $\boldsymbol{f}_i^T\boldsymbol{f}_j$ will take a small value. Thus the corresponding connection between these two points is defined as $\boldsymbol{f}_i^T\boldsymbol{f}_j$. The correct connections within each clusters are preserved in matrix $\boldsymbol{F}\boldsymbol{F}^T$, which has a clear diagonal block structure. In this paper, we use the matrix $\boldsymbol{F}\boldsymbol{F}^T$ to reconstruct the similarity matrix $\boldsymbol{W}$ as follows:

$$\min_{\boldsymbol{W}} \|\boldsymbol{W} - \boldsymbol{F}\boldsymbol{F}^T\|_F^2 \tag{5}$$

In an incremental learning model, once the new $(t+1)$-th view is obtained, the new consensus similarity matrix $\widetilde{\boldsymbol{W}}_{t+1}$ should have the following two properties: (1) It should preserve the correct connections within each clusters in the previous consensus similarity matrix $\widetilde{\boldsymbol{W}}_t$, and (2) It should preserve the correct connections within each clusters in the similarity matrix $\boldsymbol{W}^{(t+1)}$ of the new $(t+1)$-th view.

Combined with the connected graph learning, we both utilize the spectral embeddings of $\widetilde{\boldsymbol{W}}_t$ and $\boldsymbol{W}^{(t+1)}$ to reconstruct the new consensus similarity matrix $\widetilde{\boldsymbol{W}}_{t+1}$. The spectral embedding of $\widetilde{\boldsymbol{W}}_t$ is denoted as $\boldsymbol{F}_t$ and the spectral embedding of $\boldsymbol{W}^{(t+1)}$ is denoted as $\boldsymbol{F}^{(t+1)}$. Then we present the final objective function as follows:

$$\begin{aligned}\min_{\widetilde{\boldsymbol{W}}_{t+1}} \ & \mu_1\|\widetilde{\boldsymbol{W}}_{t+1} - \boldsymbol{F}_t\boldsymbol{F}_t^T\|_F^2 + \mu_2\|\widetilde{\boldsymbol{W}}_{t+1} - \boldsymbol{F}^{(t+1)}\boldsymbol{F}^{(t+1)^T}\|_F^2 \\ & + \lambda\|\widetilde{\boldsymbol{W}}_{t+1}\|_1\end{aligned} \tag{6}$$

Algorithm 1 Incremental multi-view spectral clustering with sparse and connected graph learning (SCGL)

Input: A multi-view dataset$\{\boldsymbol{X}^{(v)}\}_{v=1}^m$, cluster number c, tradeoff parameter λ.
Output: the clustering result $\boldsymbol{Y}^{(m)}$.
1: Initialize the similarity matrix $\boldsymbol{W}^{(1)}$ of the 1-th view with Eq. (1); Set the first consensus similarity matrix $\widetilde{\boldsymbol{W}}_1 = \boldsymbol{W}^{(1)}$;
for $t = 1, 2, \ldots, m-1$ **do**
2: Obtain the similarity matrix $\boldsymbol{W}^{t+1}$ of the new $(t+1)$-th view with Eq. (1);
3: Learn the spectral embedding $\boldsymbol{F}_t$ of the previous consensus similarity matrix $\widetilde{\boldsymbol{W}}_t$ with Eq. (2);
4: Learn the spectral embedding $\boldsymbol{F}^{(t+1)}$ of the similarity matrix $\boldsymbol{W}^{(t+1)}$ with Eq. (2);
while not converge **do**
5: Learn the new consensus similarity matrix $\widetilde{\boldsymbol{W}}_{t+1}$ with Eq. (10);
6: Update the weights μ_1 and μ_2 with Eq. (13);
end while
end for
Conduct spectral clustering on the consensus similarity matrix $\widetilde{\boldsymbol{W}}_m$, and obtain the final clustering result $\boldsymbol{Y}^{(m)}$.

where the first term reconstructs $\widetilde{\boldsymbol{W}}_{t+1}$ by using the spectral embedding of the previous consensus similarity matrix $\widetilde{\boldsymbol{W}}_t$. The second term reconstructs $\widetilde{\boldsymbol{W}}_{t+1}$ by using the spectral embedding of the new similarity matrix $\boldsymbol{W}^{(t+1)}$. The third term is a sparse regularization by using l_1 norm. λ is a positive tradeoff parameter which balances the reconstruction term and sparse regularization term. It is noted that the weights μ_1 and μ_2 measure the contributions of the previous views and new view to the learning result, respectively. In order to solve the stability-plasticity dilemma, an inverse distance weighting scheme (Kang et al., 2020) is adopted, where the contributions are defined by the reconstruction error of views. When the reconstruction error is large, it means that the clustering ability of the view is weak and its contribution to the final result should be small. When the reconstruction error is small, it means that the view has a strong clustering ability and should make a greater contribution to the final result.

3.3. The optimization algorithm

In this section, we propose an optimization method to solve the consensus similarity matrix $\widetilde{\boldsymbol{W}}_{t+1}$ in Eq. (6).

Solving $\widetilde{\boldsymbol{W}}_{t+1}$ when μ_1 and μ_2 are fixed. The close-form optimal solution of $\widetilde{\boldsymbol{W}}_{t+1}$ can be obtained directly and quickly, which greatly reduces the time cost during incremental multi-view clustering. For convenience, let $\boldsymbol{A} = \boldsymbol{F}_t\boldsymbol{F}_t^T$ and $\boldsymbol{B} = \boldsymbol{F}^{(t+1)}\boldsymbol{F}^{(t+1)^T}$. The Eq. (6) is expanded as:

$$\begin{aligned}\min_{\widetilde{\boldsymbol{W}}_{t+1}} &\mu_1 tr\left(\widetilde{\boldsymbol{W}}_{t+1}^T\widetilde{\boldsymbol{W}}_{(t+1)} - 2\widetilde{\boldsymbol{W}}_{t+1}^T\boldsymbol{A} + \boldsymbol{A}^T\boldsymbol{A}\right)\\ &+\mu_2 tr\left(\widetilde{\boldsymbol{W}}_{t+1}^T\widetilde{\boldsymbol{W}}_{(t+1)} - 2\widetilde{\boldsymbol{W}}_{t+1}^T\boldsymbol{B} + \boldsymbol{B}^T\boldsymbol{B}\right)\\ &+\lambda\|\widetilde{\boldsymbol{W}}_{t+1}\|_1\end{aligned} \tag{7}$$

By simple mathematical manipulation, this optimization problem can be rewritten as:

$$\begin{aligned}\min_{\widetilde{\boldsymbol{W}}_{t+1}}\ & tr(\widetilde{\boldsymbol{W}}_{t+1}^T\widetilde{\boldsymbol{W}}_{t+1}) - \frac{2}{\mu_1+\mu_2}\widetilde{\boldsymbol{W}}_{t+1}^T(\mu_1\boldsymbol{A} + \mu_2\boldsymbol{B})\\ &+\ \mu_1\boldsymbol{A}^T\boldsymbol{A} + \mu_2\boldsymbol{B}^T\boldsymbol{B} + \lambda\|\widetilde{\boldsymbol{W}}_{t+1}\|_1\end{aligned} \tag{8}$$

It is noted that $\boldsymbol{A}$ and $\boldsymbol{B}$ are fixed when we solve for $\widetilde{\boldsymbol{W}}_{t+1}$. Removing the constant term, then the above problem is equivalent to

$$\begin{aligned}\min_{\widetilde{\boldsymbol{W}}_{t+1}}\ & tr(\widetilde{\boldsymbol{W}}_{t+1}^T\widetilde{\boldsymbol{W}}_{t+1}) - \frac{2}{\mu_1+\mu_2}\widetilde{\boldsymbol{W}}_{t+1}^T(\mu_1\boldsymbol{A} + \mu_2\boldsymbol{B})\\ &+\ \frac{\lambda}{\mu_1+\mu_2}\|\widetilde{\boldsymbol{W}}_{t+1}\|_1\end{aligned} \tag{9}$$

here we constraint $\mu_1 + \mu_2 = 1$, and let $\boldsymbol{M} = \mu_1\boldsymbol{A} + \mu_2\boldsymbol{B}$, the Eq. (9) can be transformed into the following problem.

$$\min_{\widetilde{\boldsymbol{W}}_{t+1}}\ \|\widetilde{\boldsymbol{W}}_{t+1} - \boldsymbol{M}\|_F^2 + \lambda\|\widetilde{\boldsymbol{W}}_{t+1}\|_1 \tag{10}$$

As discussed in Hu, Nie, Chang et al. (2020), the optimal solution in Eq. (10) can be obtained by a soft-thresholding operator $\mathcal{S}_\mu(x)$, where the operator is defined as:

$$\mathcal{S}_\mu(x) = \begin{cases} x-\mu, & x > \mu\\ x+\mu, & x < -\mu\\ 0, & otherwise\end{cases} \tag{11}$$

The close-form optimal solution of the problem in Eq. (10) is $\widetilde{\boldsymbol{W}}_{t+1} = \mathcal{S}_{\lambda/2}(\boldsymbol{M})$. Let m_{ij} denote the (i,j)-element in matrix $\boldsymbol{M}$ and $\widetilde{w}_{ij}$ denote the (i,j)-element in matrix $\widetilde{\boldsymbol{W}}_{t+1}$, we conduct the soft-thresholding operator element-wise as follows:

$$\widetilde{w}_{ij} = \mathcal{S}_{\lambda/2}(m_{ij}) = \begin{cases} m_{ij}-\lambda/2, & m_{ij} > \lambda/2\\ m_{ij}+\lambda/2, & m_{ij} < -\lambda/2\\ 0, & otherwise\end{cases} \tag{12}$$

Solving μ_1 and μ_2 when $\widetilde{\boldsymbol{W}}_{t+1}$ is fixed. The weight μ_1 characterizes the importance of the previous views and the weight μ_2 characterizes the importance of the last view. By adopting the inverse distance weighting scheme (Kang et al., 2020), the weights of previous views and last view are measured as:

$$\mu_1 = \frac{1}{2\|\widetilde{\boldsymbol{W}}_{t+1} - \boldsymbol{A}\|_F},\ \mu_2 = \frac{1}{2\|\widetilde{\boldsymbol{W}}_{t+1} - \boldsymbol{B}\|_F} \tag{13}$$

here for comparing with the parameter λ, the weights are normalized and the sum of two weights is set to one. When the $(t+1)$-th view is arriving, we first learn the spectral embeddings $\boldsymbol{F}_t$ and $\boldsymbol{F}^{(t+1)}$ of $\widetilde{\boldsymbol{W}}_t$ and $\boldsymbol{W}^{(t+1)}$ respectively. Then, $\widetilde{\boldsymbol{W}}_{t+1}$, μ_1 and μ_2 are updated in an alternating iterative strategy until convergence. When the new $(t+2)$-th view is arriving, the $\widetilde{\boldsymbol{W}}_{t+1}$ will be used to obtain the new consensus similarity matrix $\widetilde{\boldsymbol{W}}_{t+2}$. The process is repeated until no more view is available. The detailed procedure of our method is summarized in Algorithm 1.

3.4. Computational complexity

In Algorithm 1, the main time cost consists of three parts: The first part is to obtain the similarity matrix of the new view (step 2). The second part is to learn the spectral embeddings of the previous consensus similarity matrix and the current new similarity matrix separately (steps 3–4). The third part is to learn the new consensus similarity matrix (steps 5–6).

In step 2, the computational cost of constructing the similarity matrix is $O(n^2)$, where n is the number of samples. In steps 3–4, we need to solve two eigenvalue decomposition problems on two n^2 matrices separately, which leads to $O(2n^3)$ complexity. In steps 5–6, Learning the new consensus similarity matrix requires conducting a n^2 matrix element-wise. This process leads to $O(Tn^2)$ complexity, where T is the number of iterations. When a new view is available, the complexity of our algorithm is $O(n^2 + 2n^3 + Tn^2)$. For obtaining the final consensus similarity matrix $\widetilde{\boldsymbol{W}}_m$, the total complexity of our algorithm is $O(m(n^2 + 2n^3 + Tn^2))$.

3.5. Connection with previous studies

The main task of incremental multi-view clustering is to fuse a collection of views one by one. In Zhou et al. (2019), a set of base kernels constructed from the previous t views are preserved. When the new view is available, the set of base kernels and the new kernel constructed from the new view are combined to learn a consensus kernel. Although this approach is consistent with the task of our approach, a set of basic kernels needs to be updated each time when a new view is available. This approach is time-consuming and difficult to implement. In this paper, we only preserve a similarity matrix to represent the structure information of all the previous views which greatly reduces the complexity of our method.

In addition, in order to improve clustering performance, many multi-view clustering methods need to constrain the rank of similarity matrix, such as Zhan et al. (2018) and Nie, Cai, and Li (2017). Since the data graph has exactly c connected components and c is the number of clusters, the rank of the associated similarity matrix is constrained to c in these methods. In this paper, we utilize a matrix $\boldsymbol{F}\boldsymbol{F}^T$ to reconstruct the similarity matrix as in Eq. (5). Because $\boldsymbol{F} \in \mathcal{R}^{n\times c}$, we have:

$$rank(\boldsymbol{F}\boldsymbol{F}^T) \le rank(\boldsymbol{F}) \le \min(n, c) \tag{14}$$

here n is the number of samples and c is the number of clusters, $c \ll n$. Therefore, the rank of similarity matrix in our method is no more than c. This property indicates that our reconstruction term maintains good connectivity of similarity matrix from another perspective.

4. Experiments

In the experimental part, we verify our proposed method SCGL on six real world datasets, where three widely used cluster evaluation metrics are utilized including clustering accuracy (ACC), normalized mutual information (NMI) and adjusted rand index (ARI). All experiments are conducted on a HP computer with Intel Core i5-8500 CPU (3.00 GHZ, 8G RAM).

4.1. Datasets

The experiments are performed on six widely used real world datasets. All datasets are summarized in Table 2.

Table 2
Multi-view datasets.

Datasets	♯ Samples	♯ Clusters	♯ Views
MSRC	210	7	5
BBC	685	5	4
100leaves	1600	100	3
Handwritten	2000	10	6
Out-Scene	2688	8	4
Caltech101	9144	102	6

MSRC dataset (Hu, Nie, Chang et al., 2020) contains seven classes of scene recognition images with a total of 210 samples. Each sample is described by five features including the 24 dimension color moment (View 1), the 576 dimension HOG (View 2), the 512 dimension GIST (View 3), the 256 dimension LBP (View 4), and the 254 dimension Centrist feature (View 5).

BBC dataset[1] contains 685 document samples of five topical labels. These documents are split into four segments, where each segment can be treated as a view. The first view is the 4659 dimension, the second is the 4633 dimension, the third view is the 4665 dimension, and the fourth view is the 4684 dimension.

100leaves dataset[2] contains one hundred plant species with a total of 1600 samples. Each sample has three views. The first view is the 64 dimension shape descriptor, the second is the 64 dimension fine scale margin, and the third is the 64 dimension texture histogram.

Handwritten dataset[3] contains ten types of handwritten numbers from 0 to 9 with a total of 2000 samples. Each sample is described by six features, each of which is treated as a view. The first feature is the 240 dimension pixel averages in 2×3 windows, the second is the 76 dimension Fourier coefficients, the third is the 216 dimension profile correlations, the fourth is the 47 dimension Zernike moment, the fifth is the 64 dimension Karhunen–Love coefficients, and the sixth is the 6 dimension morphological features.

Out-Scene dataset[4] consists of eight groups with a total of 2688 image samples. For each sample, it is described by four different features including the 512 dimension GIST (View 1), the 432 dimension color moment (View 2), the 256 dimension HOG (View 3), and the 48 dimension LBP (View 4).

Caltech101 dataset[5] contains 9144 image samples belonging to 102 classes. There are six different features for each sample including the 48 dimension Gabor feature (View 1), the 40 dimension wavelet-moment feature (View 2), the 254 dimension Centrist feature (View 3), the 1984 dimension HOG feature (View 4), the 512 dimension GIST feature (View 5), and the 928 dimension LBP feature (View 6).

4.2. Comparison methods

For evaluating the performance of our method, we compare it with the classic single view spectral clustering (von Luxburg, 2007) and six state-of-the-art multi-view clustering methods, including CorSC (Kumar et al., 2011), AWP (Nie, Tian, & Li, 2018), MLAN (Nie et al., 2017), NESE (Hu, Nie, Wang et al., 2020), SMVSC (Hu, Nie, Chang et al., 2020), IMSC (Zhou et al., 2019).

Spectral clustering (SC) is a classical single view clustering method, which is conducted on each view as the baseline.

Co-regularized multi-view spectral clustering (CorSC) assumes that the corresponding samples in each view have the same cluster membership. It learns a common spectral embedding by co-regularizing the different spectral embeddings of different views.

Multiview clustering via adaptively weighted procrustes (AWP) is a parameter-free clustering model by considering the clustering capacity of different views. AWP extends the spectral rotation to multi-view clustering, thus eliminating the instability of subsequent clustering.

Multi-view learning with adaptive neighbors (MLAN) belongs to graph-based multi-view clustering. MLAN performs structure graph learning and multi-view clustering simultaneously. Specifically, MLAN optimizes the structure of the graph by constraining the rank of the Laplace matrix.

Multi-view spectral clustering via integrating nonnegative embedding and spectral embedding (NESE) inherits the advantages of both NMF-based methods and graph-based methods. The cluster indicators are directly revealed by the nonnegative embedding, where the uncertainty brought by post-processing can be avoided.

Multi-view spectral clustering via sparse graph learning (SMVSC) also belongs to graph-based multi-view clustering. It learns a consistent similarity matrix with a l_1 regularization from multiple views. The most obvious advantage is that the consistent similarity matrix can be solved directly without iterative optimization.

Incremental multi-view spectral clustering (IMSC) also adopts an incremental strategy that is similar to our method. It constructs a few base kernels and a spectral embedding from the first few views. When a new view is arriving, IMSC learns a consensus kernel from the base kernels and the new view. Meanwhile, the base kernels are updated. The current new spectral embedding is obtained by joining the consensus kernel and the previous spectral embedding.

4.3. Parameter selection

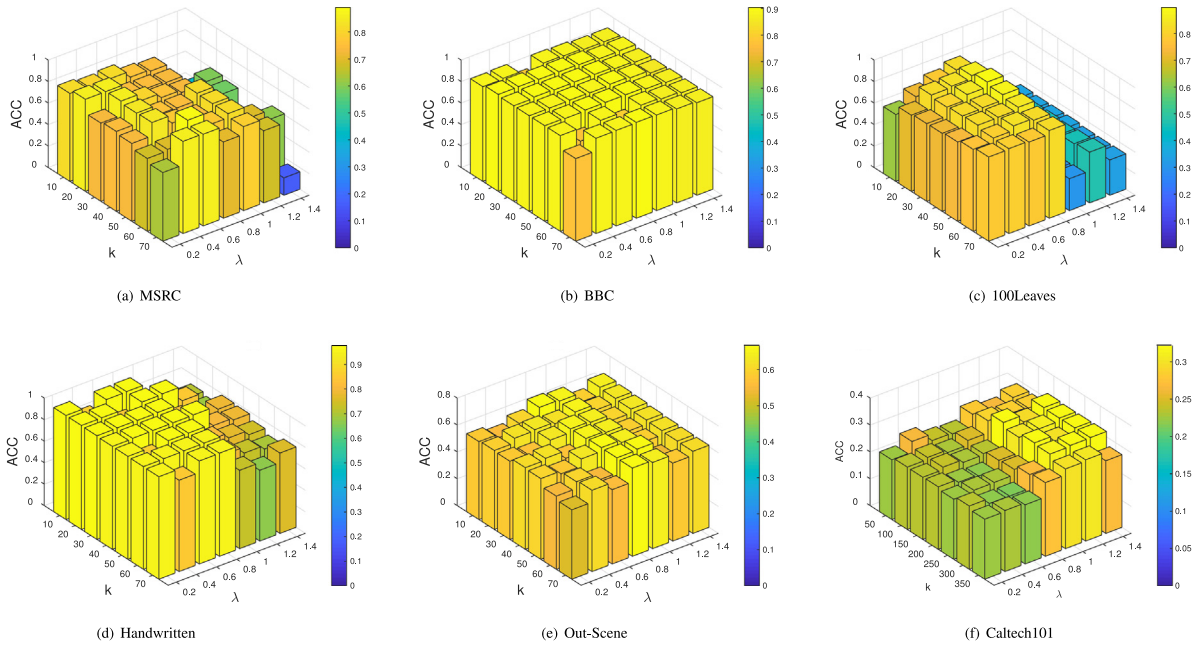
In our model, there are two parameters: the number of neighbors k and the tradeoff parameter λ. The former affects the construction of KNN adjacent matrix, and the latter affects the effect of l_1 regularization. Thus, we validate different combinations of parameters by grid searching, where k is varied in $\{10, 20, 30, 40, 50, 60, 70\}$ and λ is varied in $\{0.2, 0.4, 0.6, 0.8, 1.0, 1.2, 1.4\}$. Note that for Caltech101, the scale is much larger. It needs more neighbors to construct the similarity matrix, where k is varied in $\{50, 100, 150, 200, 250, 300, 350\}$. The results on six datasets with different combinations of k and λ are shown in Fig. 2.

In Fig. 2, we can see that λ and k both have a great influence on the experimental results. It should be noted that the main purpose of sparsity constraint is to reduce false connections between points by alleviating noise. Therefore, too small λ will lead to insufficient sparsity, which will fail to alleviate noise. Conversely, too large λ will lead to excessive sparsity, which will sacrifice connectivity. From the experiment results in Fig. 2, the optimal parameters of our method on each dataset are set as follows: MSRC ($k = 60, \lambda = 0.6$), BBC ($k = 20, \lambda = 0.4$), 100leaves ($k = 30, \lambda = 0.8$), Handwritten ($k = 20, \lambda = 0.8$), Out-Scene ($k = 60, \lambda = 1$), Caltech101 ($k = 150, \lambda = 1$).

4.4. Comparison with other methods

Each algorithm contains parameters that are set according to their paper. In CorSC, the weight parameter specific to vth view is set to $1/m$, where m is the number of views. In NESE, the weight

[1] http://mlg.ucd.ie/datasets/segment.html.
[2] https://archive.ics.uci.edu/ml/datasets/One-hundred+plant+species+leaves+data+set.
[3] http://archive.ics.uci.edu/ml/datasets/Multiple+Features.
[4] https://github.com/sudalvxin/SMSC/tree/master/data.
[5] http://www.vision.caltech.edu/ImageDatasets/Caltech101/.



(a) MSRC (b) BBC (c) 100Leaves

(d) Handwritten (e) Out-Scene (f) Caltech101

Fig. 2. The experiment results of SCGL on six real world datasets with different combinations of the number of neighbors k and the tradeoff parameter λ.

Table 3
The experimental results on MSRC dataset. The best results are highlighted in bold.

Views	Metric	SC	CorSC	AWP	MLAN	NESE	SMVSC	IMSC	SCGL
2	ACC	0.6366	0.5052	**0.7619**	0.7285	0.6191	0.1476	0.5666	0.7571
	NMI	0.5251	0.3764	**0.6745**	0.6378	0.5274	0.0299	0.4589	0.6516
3	ACC	0.6401	0.6357	0.7047	0.6714	0.7619	0.4333	0.6666	**0.8285**
	NMI	0.5707	0.5475	0.6613	0.6132	0.7076	0.3559	0.5324	**0.7485**
4	ACC	0.4609	0.6504	0.8142	0.6904	0.8047	0.8233	0.6333	**0.8333**
	NMI	0.4113	0.5846	**0.7599**	0.6584	0.7321	0.7376	0.5235	0.7402
5	ACC	0.5309	0.6881	0.8001	0.6809	0.8142	0.8761	0.6619	**0.8904**
	NMI	0.4621	0.6099	0.7166	0.6299	0.7230	0.7858	0.6191	**0.7936**

Table 4
The experimental results on BBC dataset. The best results are highlighted in bold.

Views	Metric	SC	CorSC	AWP	MLAN	NESE	SMVSC	IMSC	SCGL
2	ACC	0.5216	0.6358	0.8291	0.6817	**0.8891**	0.5503	0.4277	0.7868
	NMI	0.3341	0.5242	0.6719	0.4648	**0.7181**	0.3081	0.1819	0.6610
3	ACC	0.5608	0.6274	0.7085	0.4525	**0.8934**	0.7868	0.4481	0.8846
	NMI	0.4028	0.4859	0.5614	0.1903	**0.7377**	0.6279	0.1856	0.7241
4	ACC	0.5551	0.6482	0.8765	0.6875	0.8921	0.7927	0.4467	**0.9036**
	NMI	0.3362	0.5443	0.7198	0.4885	0.7487	0.7161	0.1868	**0.7533**

Table 5
The experimental results on 100Leaves dataset. The best results are highlighted in bold.

Views	Metric	SC	CorSC	AWP	MLAN	NESE	SMVSC	IMSC	SCGL
2	ACC	0.4371	0.6475	0.7106	0.7325	0.6762	0.4218	**0.8156**	0.7275
	NMI	0.7029	0.8281	0.8468	0.8716	0.8174	0.5703	**0.8809**	0.8563
3	ACC	0.5165	0.8012	0.8087	0.8662	0.8655	**0.8968**	0.8852	0.8806
	NMI	0.7415	0.9226	0.9061	0.9407	0.9218	**0.9576**	0.9386	0.9458

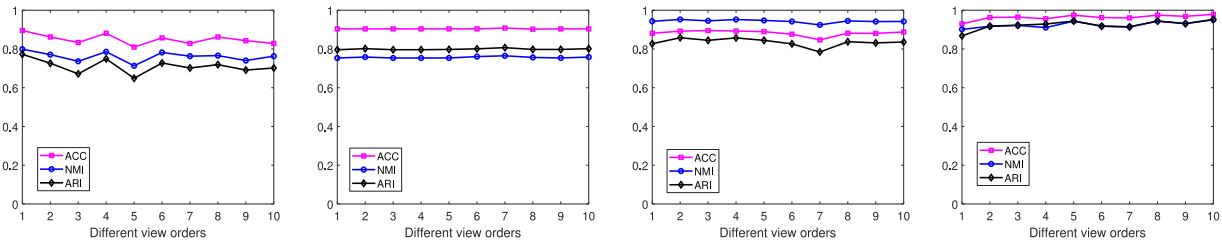
Table 6
The experimental results on Handwritten dataset. The best results are highlighted in bold.

Views	Metric	SC	CorSC	AWP	MLAN	NESE	SMVSC	IMSC	SCGL
2	ACC	0.6605	0.8611	0.9701	0.9732	0.8645	0.8435	0.9524	**0.9765**
	NMI	0.6329	0.7779	0.9313	0.9339	0.8633	0.8373	0.9022	**0.9464**
3	ACC	0.6132	0.8181	0.7725	0.9745	0.7542	0.7333	0.9515	**0.9755**
	NMI	0.6134	0.7727	0.7985	0.9409	0.7892	0.8307	0.9027	**0.9434**
4	ACC	0.5353	0.7591	0.8625	**0.9755**	0.7515	0.8512	0.9452	0.9771
	NMI	0.4851	0.7287	0.8531	0.9436	0.7809	0.8922	0.9035	**0.9464**
5	ACC	0.6951	0.7728	0.8681	0.9721	0.8735	0.8346	0.9635	**0.9765**
	NMI	0.6461	0.7352	0.8637	0.9372	0.8744	0.8752	0.9221	**0.9454**
6	ACC	0.4506	0.8148	0.8681	**0.9721**	0.8675	0.8498	0.9691	0.9720
	NMI	0.4799	0.7751	0.8636	**0.9372**	0.8711	0.9021	0.9307	0.9364

parameter is independently determined by the reconstruction error of each view. There is no weight parameter explicitly defined in AWP, MLAN and SMVSC. In MLAN, the regularization parameter α is determined by adaptive neighbors. In SMVSC, the optimal parameter for sparse term is selected from the set $\{0.02, 0.06, 0.1, 0.14, 0.18, 0.22\}$. In IMSC, the balancing parameters λ_1, λ_2 and γ are set to 1, 1 and 0.1, respectively. In Section 4.3, the optimal parameter of our method has been obtained. It is noted that SC, CorSC, AWP, NESE, SMVSC and our SCGL initialize the similarity matrices of each view by using KNN adjacency matrix. For fair comparison, the number of neighbors is consistent with our method in Section 4.3. In MLAN, the neighbors to each of samples are assigned adaptively. In IMSC, the kernel matrix is obtained by using the Random Fourier features method.

In Tables 3–8, the SC is conducted separately on each view. When a new view is arriving, traditional multi-view clustering methods are conducted on all collected views. In our method, we construct the previous graph using the first view and start incremental multi-view clustering by fusing the second view. When a new view is arriving, the consensus similarity matrix is updated by using its previous version and the similarity matrix of the new view. The experimental results in terms of ACC, NMI show the following points:

1. None of methods performs best on all datasets. We observed that our method performed well on most datasets. Although our method is not the best on some datasets, the difference between the best results is very small, which proves that our method can accomplish the clustering task of multi-view data well.
2. When the number of views is increasing, the performance of our method is gradually improved. In particular, our



(a) MSRC (b) BBC (c) 100Leaves (d) Handwritten

Fig. 3. The experiment results of SCGL in ten different view orders.

Table 7
The experimental results on Out-Scene dataset. The best results are highlighted in bold.

Views	Metric	SC	CorSC	AWP	MLAN	NESE	SMVSC	IMSC	SCGL
2	ACC	0.5181	0.6228	0.6328	0.4773	0.6067	0.1569	**0.6656**	0.6227
	NMI	0.3935	0.4916	0.5064	0.3991	0.4774	0.0118	**0.5406**	0.5328
3	ACC	0.3381	0.5999	0.5844	0.5334	0.6171	**0.6701**	0.6343	0.6573
	NMI	0.1785	0.4615	0.5001	0.4605	0.5034	**0.5968**	0.5447	0.5265
4	ACC	0.2446	0.6179	0.6123	0.5364	0.6101	0.5587	0.6373	**0.6557**
	NMI	0.0791	0.5026	0.5102	0.4589	0.4994	0.5362	0.5110	**0.5487**

Table 8
The experimental results on Caltech101 dataset. The best results are highlighted in bold.

Views	Metric	SC	CorSC	AWP	MLAN	NESE	SMVSC	IMSC	SCGL
2	ACC	0.1389	0.1147	0.1268	0.1193	0.1327	0.1006	0.1251	**0.2101**
	NMI	0.3202	0.2679	0.2991	0.1573	0.3029	0.0442	0.3138	**0.3284**
3	ACC	0.1401	0.1280	0.1578	0.1283	0.1629	0.1523	0.1360	**0.2555**
	NMI	0.3234	0.2959	0.3305	0.1649	0.3381	0.2021	**0.3292**	0.2995
4	ACC	0.2667	0.1627	0.2630	0.2014	0.2111	0.3173	0.1615	**0.3325**
	NMI	0.4839	0.3526	0.5032	0.2885	0.4088	0.4570	0.3591	**0.4952**
5	ACC	0.2581	0.1945	0.2807	0.2218	0.2577	**0.3283**	0.1741	0.3193
	NMI	0.4609	0.3927	**0.5165**	0.3147	0.4577	0.5077	0.3889	0.5069
6	ACC	0.2052	0.2052	0.2818	0.2021	0.2589	0.3268	0.1816	**0.3343**
	NMI	0.4241	0.4096	**0.5119**	0.2756	0.4721	0.5192	0.4034	0.5036

method performs better than IMSC, where IMSC is a typical incremental multi-view clustering method. This phenomenon proves that our approach is better at incrementally learning new views. In order to prove whether our method is sensitive to views order, we conduct further test in Section 4.5.

3. It can be seen from the results of SC that some views have good clustering effect, while some views have poor clustering effect. When fusing the bad views, most multi-view clustering method perform unsatisfactory. The main reason is that these methods do not preserve the correct connections within each clusters during the incremental process. This phenomenon proves the effectiveness of the connected graph reconstruction introduced in our method.
4. Although our method and SMVSC both adopt sparse graph learning, our method performs better than SMVSC on most datasets. This phenomenon is due to that our method not only uses the sparse graph learning, but also combines the connected graph learning. The l_1 regularization has a great effect in our model. This is mainly because too small lambda should prevent noise from being removed, and too large lambda will sacrifice a lot of correct connections within clusters. Therefore, the connected graph learning should be incorporated into our model, which can preserve the correct connections within clusters.

Table 9
Ablation study on six real world datasets.

Dataset	Metric	CGL	SGL	SCGL
MSRC	ACC	0.74±0.01	0.79±0.05	0.89±0.01
	NMI	0.62±0.01	0.71±0.03	0.79±0.01
	ARI	0.54±0.01	0.65±0.04	0.76±0.00
BBC	ACC	0.83±0.03	0.85±0.07	0.90±0.00
	NMI	0.71±0.04	0.73±0.04	0.75±0.00
	ARI	0.72±0.04	0.75±0.06	0.79±0.00
100leaves	ACC	0.66±0.01	0.86±0.02	0.87±0.01
	NMI	0.82±0.00	0.94±0.00	0.94±0.00
	ARI	0.55±0.01	0.81±0.02	0.82±0.00
Handwritten	ACC	0.61±0.03	0.75±0.04	0.97±0.00
	NMI	0.61±0.02	0.80±0.03	0.93±0.00
	ARI	0.47±0.04	0.69±0.04	0.93±0.00
Out-Scene	ACC	0.39±0.00	0.55±0.03	0.64±0.00
	NMI	0.24±0.00	0.50±0.02	0.54±0.00
	ARI	0.16±0.00	0.40±0.01	0.47±0.00
Caltech101	ACC	0.21±0.01	0.11±0.00	0.32±0.03
	NMI	0.44±0.01	0.10±0.00	0.48±0.02
	ARI	0.15±0.01	0.09±0.00	0.26±0.02

4.5. Experiment on the effect of view order

Since the consensus similarity graph is built incrementally using the arriving views, we provide an experimental study on the effect of view order on the final clustering. In Fig. 3, the horizontal axis represents the different view orders and the vertical axis shows the clustering results in 10 different view orders. The first view order is the original order. The remaining nine orders are randomly generated. Fig. 3 shows that the clustering results of our method fluctuate very little in different view orders. This phenomenon proves that our method is not sensitive to the order of views.

4.6. Ablation study

In order to show the importance of each term in Eq. (6), we conduct ablation study by comparing our method with two baselines. The two baselines are the special cases of our method, which only consider one component of our SCGL. One baseline removes the l_1 regularization and only adopts the connected graph learning (CGL). The other baseline removes the reconstruction term and only adopts the sparse graph learning (SGL), where the objective function is defined in Eq. (4). The clustering results in terms of ACC, NMI and ARI are shown in Table 9.

According to the results reported in Table 9, we can see that SCGL performs better than CGL on all datasets. This phenomenon proves that the sparse graph learning can improve the multi-view clustering ability. On the other hand, we can see that SCGL also performs better than SGL, which proves that the connected graph learning is important to our model as well. Therefore, joint sparse graph learning and connected graph learning can better improve the effect of incremental multi-view clustering.

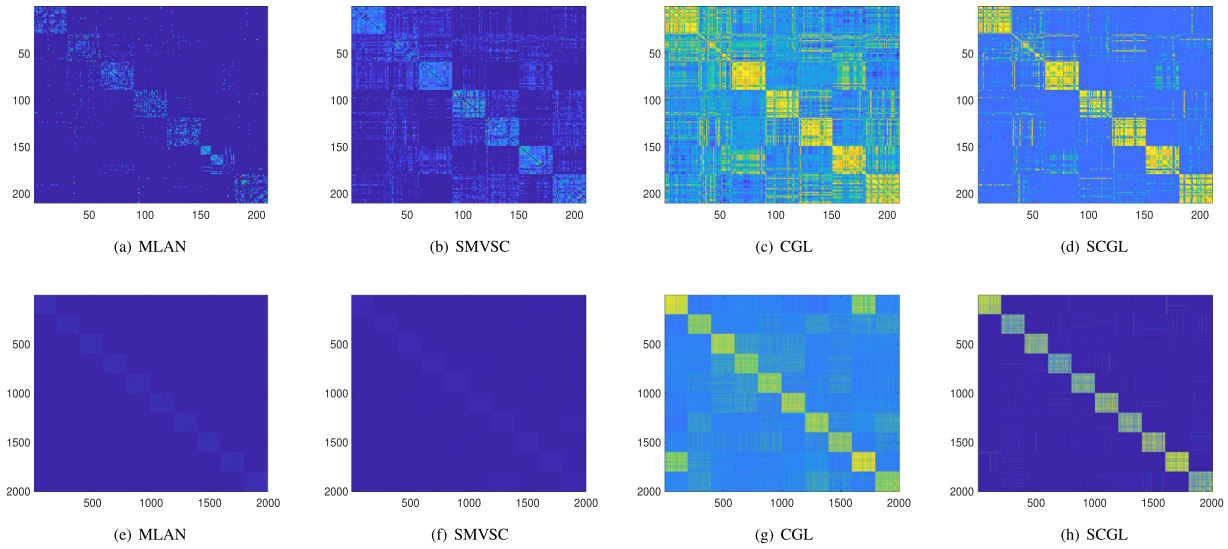


Fig. 4. Visualization of the learned similarity matrix by using MLAN, SMVSC, CGL and SCGL. The top row is the result on MSRC dataset, and the bottom row is the result on Handwritten dataset.

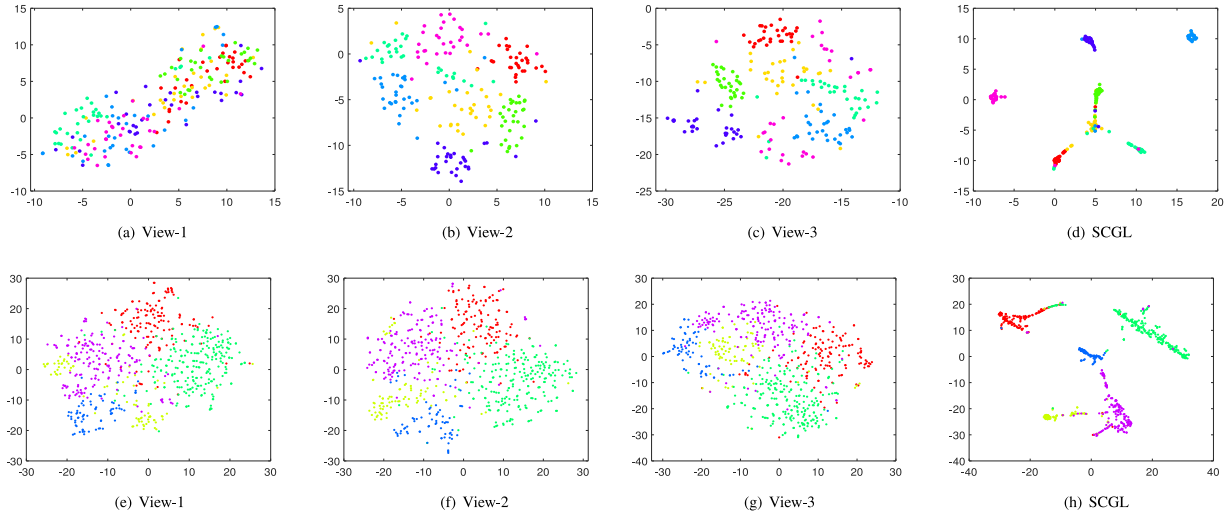


Fig. 5. Visualization of the clustering results by using t-SNE. The top row is the result on MSRC dataset, and the bottom row is the result on BBC dataset.

4.7. Visualization

In our method, the consensus similarity matrix reveals the structure of multi-view data, which determines the performance of clustering task. In Fig. 4, the learned similarity matrices in four different methods are visualized, including MLAN, SMVSC, CGL and SCGL. In MLAN, the similarity matrices are obtained by adding rank constraint without considering sparsity and connectivity. In SMVSC, only sparsity is considered when obtaining the similarity matrices. In CGL, only connectivity is considered when obtaining the similarity matrices. From Fig. 4, we can observe that the similarity matrix of our method SCGL not only contains less noise, but also has a clearer diagonal block structure. This is mainly because our method SCGL incorporates the sparsity and the connectivity simultaneously to learn the consensus similarity matrix.

In order to further verify the superiority of our method, the clustering results of our method are visualized by using classic t-SNE method (Hinton, 2008). In Fig. 5, the internal structure of the original data of different views are visualized respectively. It can be seen that in the original data, not only the distance between different groups is small, but also there are overlaps between different groups. In SCGL, the boundaries between different groups is very clear, and there is basically no overlaps between different groups. The visualization results further demonstrate the superiority of our method.

4.8. Time cost

Table 10 shows the running time for learning the clustering result in the streaming view setting. When a new view is arriving, the compared methods are conducted on all collected views, except IMSC. Both IMSC and SCGL incrementally learn the clustering results. When a new view is arriving, the times of IMSC and SCGL used for incremental learning the current clustering results are reported.

In Table 10, it can be seen that the time cost of SCGL is less than other methods, except SMVSC. SMVSC is fast because it does not assign weights to each view. Therefore, there is no iteration to obtain the optimal adaptive weights in SMVSC. In addition, we observed that SCGL is faster than IMSC as the number of views increases. The main reason is that IMSC needs to constantly update multiple base kernel matrices. In contrast, SCGL requires only one

Table 10
Running time (in seconds) on Handwritten and Out-Scene datasets.

Algorithm	Handwritten					Algorithm	Out-Scene		
	2 views	3 views	4 views	5 views	6 views		2 views	3 views	4 views
CorSC	8.743	11.135	13.722	17.665	21.313	CorSC	33.375	45.366	54.781
AWP	2.043	3.115	4.293	5.865	7.896	AWP	3.105	4.836	6.771
MLAN	19.079	19.751	17.434	19.527	20.719	MLAN	51.404	48.093	56.799
NESE	6.386	9.979	13.743	17.543	24.028	NESE	10.495	16.727	20.785
SMVSC	1.215	1.236	1.531	1.618	1.875	SMVSC	2.773	2.335	2.576
IMSC	0.947	1.438	1.737	2.116	2.206	IMSC	1.803	3.078	3.365
SCGL	3.466	1.243	1.218	1.298	2.035	SCGL	4.912	2.523	2.498

consensus similarity matrix to be updated. The results of time analysis prove that SCGL is efficient for multi-view clustering when the number of views is increasing.

5. Conclusion

In this paper, we proposed a novel incremental multi-view spectral clustering with sparse and connected graph learning. Compared with the traditional multi-view clustering methods, our method can incrementally fuse newly collected views by updating the consensus similarity matrix. In order to improve the clustering performance, the sparse graph learning and the connected graph learning are integrated into our model, which can not only reduce the noises, but also preserve the correct connections within clusters. The inverse distance weighting scheme is adopted in our model to solve the stability-plasticity dilemma. The experimental results demonstrate that our method has excellent multi-view clustering performance when the number of views is increasing. Meanwhile, our method is not sensitive to the order of views, which makes it suitable to deal with the newly collected views.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Funding

This work is partially supported by National Natural Science Foundation of China (61772198, 62072151), Zhejiang Basic Public Welfare Research Project, China (LGN18F020002), Natural Science Foundation of Zhejiang Province, China (LR20F020002), Anhui Provincial Natural Science Fund for Distinguished Young Scholars, China (2008085J30), the Fundamental Research Funds for Central Universities of China (JZ2019HGPA0102), and the joint funds of the National Natural Science Foundation of China (U20A20228).

Ethical approval

This article does not contain any studies with human participants or animals performed by any of the authors.

References

Fu, L., Lin, P., Vasilakos, A. V., & Wang, S. (2020). An overview of recent multi-view clustering. *Neurocomputing*, *402*, 148–161.

Gao, H., Nie, F., Li, X., & Huang, H. (2015). Multi-view subspace clustering. In *IEEE international conference on computer vision* (pp. 4238–4246).

Gao, Q., Wan, Z., Liang, Y., Wang, Q., Liu, Y., & Shao, L. (2020). Multi-view projected clustering with graph learning. *Neural Networks*, *126*, 335–346.

Gao, S., Yu, Z., Jin, T., & Yin, M. (2019). Multi-view low-rank matrix factorization using multiple manifold regularization. *Neurocomputing*, *335*, 143–152.

He, H., Chen, S., Li, K., & Xu, X. (2011). Incremental learning from stream data. *IEEE Transactions Neural Networks*, *22*(12), 1901–1914.

Hinton, G. E. (2008). Visualizing high-dimensional data using t-SNE. *Journal of Machine Learning Research*, *9*(2), 2579–2605.

Hu, Z., Nie, F., Chang, W., Hao, S., Wang, R., & Li, X. (2020). Multi-view spectral clustering via sparse graph learning. *Neurocomputing*, *384*, 1–10.

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

Huang, S., Xu, Z., Tsang, I. W., & Kang, Z. (2020). Auto-weighted multi-view co-clustering with bipartite graphs. *Information Sciences*, *512*, 18–30.

Kang, Z., Shi, G., Huang, S., Chen, W., Pu, X., Zhou, J. T., et al. (2020). Multi-graph fusion for multi-view spectral clustering. *Knowledge-Based Systems*, *189*, Article 105102.

Kumar, A., & III, H. D. (2011). A co-training approach for multi-view spectral clustering. In L. Getoor, & T. Scheffer (Eds.), *Proceedings of the 28th international conference on machine learning* (pp. 393–400).

Kumar, A., Rai, P., & III, H. D. (2011). Co-regularized multi-view spectral clustering. In *Advances in neural information processing systems 24: 25th annual conference on neural information processing systems* (pp. 1413–1421).

Li, Z., Tang, C., Chen, J., Wan, C., Yan, W., & Liu, X. (2019). Diversity and consistency learning guided spectral embedding for multi-view clustering. *Neurocomputing*, *370*, 128–139.

Li, Z., Wang, Q., Tao, Z., Gao, Q., & Yang, Z. (2019). Deep adversarial multi-view clustering network. In *Proceedings of the 28th international joint conference on artificial intelligence* (pp. 2952–2958).

Li, J., Zhou, G., Qiu, Y., Wang, Y., Zhang, Y., & Xie, S. (2020). Deep graph regularized non-negative matrix factorization for multi-view clustering. *Neurocomputing*, *390*, 108–116.

Liu, H., & Ban, X. (2015). Clustering by growing incremental self-organizing neural network. *Expert Systems with Applications*, *42*(11), 4965–4981.

Liu, J., Wang, C., Gao, J., & Han, J. (2013). Multi-view clustering via joint nonnegative matrix factorization. In *Proceedings of the 13th SIAM international conference on data mining* (pp. 252–260).

Mehrkanoon, S., Agudelo, O. M., & Suykens, J. A. K. (2015). Incremental multi-class semi-supervised clustering regularized by Kalman filtering. *Neural Networks*, *71*, 88–104.

Miao, M., Zhang, W., Hu, W., & Wang, R. (2020). An adaptive multi-domain feature joint optimization framework based on composite kernels and ant colony optimization for motor imagery EEG classification. *Biomedical Signal Processing and Control*, *61*, Article 101994.

Nie, F., Cai, G., & Li, X. (2017). Multi-view clustering and semi-supervised classification with adaptive neighbours. In *Proceedings of the 31th AAAI conference on artificial intelligence* (pp. 2408–2414).

Nie, F., Tian, L., & Li, X. (2018). Multiview clustering via adaptively weighted procrustes. In *Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining* (pp. 2022–2030).

Rosenfeld, A., & Tsotsos, J. K. (2020). Incremental learning through deep adaptation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *42*(3), 651–663.

Roy, D., Panda, P., & Roy, K. (2020). Tree-CNN: A hierarchical deep convolutional neural network for incremental learning. *Neural Networks*, *121*, 148–160.

von Luxburg, U. (2007). A tutorial on spectral clustering. *Statistics and Computing*, *17*(4), 395–416.

Wang, X., Lei, Z., Guo, X., & Zhang, C. (2019). Multi-view subspace clustering with intactness-aware similarity. *Pattern Recognition*, *88*, 50–63.

Wang, H., Wang, Y., Zhang, Z., Fu, X., & Wang, M. (2020). Kernelized multiview subspace analysis by self-weighted learning. *IEEE Transactions on Multimedia*.

Wang, F., Zhu, L., Liang, C., Li, J., Chang, X., & Lu, K. (2020). Robust optimal graph clustering. *Neurocomputing*, *378*, 153–165.

Wen, J., Zhang, Z., Zhang, Z., Fei, L., & Wang, M. (2021). Generalized incomplete multiview clustering with flexible locality structure diffusion. *IEEE Transactions on Cybernetics*, *51*(1), 101–114.

Wen, J., Zhang, Z., Zhang, Z., Wu, Z., Fei, L., Xu, Y., et al. (2020). DIMC-net: deep incomplete multi-view clustering network. In *Proceedings of the 28th ACM international conference on multimedia* (pp. 3753–3761).

Wen, J., Zhang, Z., Zhang, Z., Zhu, L., Fei, L., Zhang, B., et al. (2020). Unified tensor framework for incomplete multi-view clustering and missing-view inferring. In *Proceedings of the 35th AAAI Conference on artificial intelligence AAAI, virtual conference*.

Yin, H., Li, F., Zhang, L., & Zhang, Z. (2019). Multi-view clustering via spectral embedding fusion. *Soft Computing*, *23*(1), 343–356.

Zhan, K., Zhang, C., Guan, J., & Wang, J. (2018). Graph learning for multiview clustering. *IEEE Transactions on Cybernetics*, *48*(10), 2887–2895.

Zhang, Z., Liu, L., Shen, F., Shen, H. T., & Shao, L. (2019). Binary multi-view clustering. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *41*(7), 1774–1782.

Zhang, X., Zhao, L., Zong, L., Liu, X., & Yu, H. (2014). Multi-view clustering via multi-manifold regularized nonnegative matrix factorization. In *2014 IEEE international conference on data mining* (pp. 1103–1108). IEEE Computer Society.

Zhou, P., Shen, Y., Du, L., Ye, F., & Li, X. (2019). Incremental multi-view spectral clustering. *Knowledge-Based Systems*, *174*, 73–86.

Zhu, W., Lu, J., & Zhou, J. (2019). Structured general and specific multi-view subspace clustering. *Pattern Recognition*, *93*, 392–403.

Zong, L., Zhang, X., Zhao, L., Yu, H., & Zhao, Q. (2017). Multi-view clustering via multi-manifold regularized nonnegative matrix factorization. *Neural Networks*, *88*, 74–89.