

One-step Kernel Multi-view Subspace Clustering[☆]

Guang-Yu Zhang^a, Yu-Ren Zhou^{a,b,*}, Xiao-Yu He^a, Chang-Dong Wang^a, Dong Huang^c

^a School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China

^b Engineering Research Institute, Guangzhou College of South China University of Technology, Guangzhou, China

^c College of Mathematics and Informatics, South China Agricultural University, Guangzhou, China

ARTICLE INFO

Article history:

Received 11 June 2019

Received in revised form 11 October 2019

Accepted 12 October 2019

Available online 17 October 2019

Keywords:

Subspace clustering

Multi-view data

One-step strategy

Kernelized model

Simplex constraint

Rank constraint

ABSTRACT

Multi-view subspace clustering is essential to many scientific problems. However, most existing methods suffer from three aspects of issues. First, these methods usually adopt a two-step framework, lacking the ability to achieve an optimal common affinity matrix across multiple views. Second, these methods are intended to solve the clustering problem in linear subspaces but may fail in practice as most real-world data sets may exhibit non-linear structures. Third, most existing subspace-based methods force the negative elements in the coefficient matrix to be positive, which may damage the inherent correlation among the data. To address above issues, we propose a novel approach termed One-step Kernel Multi-view Subspace Clustering (OKMSC). The common affinity matrix is learned from all views under one-step framework, which integrates the nonnegative and discriminative property of affinity matrix into the computation. Further, a kernelized model is designed to address the nonlinear multi-view clustering problem. And an iterative optimization method is designed to solve the objective function in this model. Extensive experiments have validated the superiority of the proposed method over several state-of-art clustering methods.

© 2019 Elsevier B.V. All rights reserved.

1. Introduction

Multi-view data are commonly observed in a variety of data mining and computer vision problems. For example, in face clustering, the same human face can be captured by different devices, which are taken as multiple views; When it comes to document clustering, one document can be translated into multiple languages and each language is referred to one separate view; For web page classification, a web page can be described by three views, such as text, image and hyperlinks. Generally, multi-view data contain complementary information while different views can describe distinct perspectives of data. The key problem for multi-view data analysis resides in how to integrate the heterogeneous information from multiple views while discovering the latent structures. Multi-view data analysis has drawn increasing attention over the past years and an amount of multi-view learning methods have been proposed in various tasks, remarkable

examples include multi-view subspace learning [1], multi-view collaborative learning [2], multi-view dimension reduction [3], multi-view feature selection [4], multi-view clustering [5,6], etc. In this paper, we focus on the problem of multi-view clustering.

Recently, a range of multi-view clustering methods have been extensively studied in the literatures [7–10]. In general, they can be categorized into four classes: *k*-means based methods [11,12], matrix factorization based methods [13,14], graph-based methods [15,16] and subspace-based methods [17]. Among these four methods, graph-based methods and subspace-based methods have become the mainstream due to their effectiveness and theoretical guarantees. The graph-based methods are designed based on the spectral model, which exploits the consistency among different views by graph fuse strategy. They generally follow a two-step framework to learn the common graph: (1) construct the affinity matrix in each view; (2) fuse multiple affinity matrices into one common affinity matrix. Typical graph-based methods include Auto-weighted Multiple Graph Learning (AMGL) [18] and Self-weighted Multi-view Clustering with multiple graphs (SwMC) [19]. Although graph-based methods have achieved promising performance in some scenario, their performance is highly dependent on the quality of pre-defined affinity matrix in each view. The final clustering performance of these methods is also influenced by many factors in the setting of pre-defined affinity matrix, such as the type of affinity matrix, the neighborhood size, the noisy and outliers.

[☆] No author associated with this paper has disclosed any potential or pertinent conflicts which may be perceived to have impending conflict with this work. For full disclosure statements refer to <https://doi.org/10.1016/j.knosys.2019.105126>.

* Corresponding author at: School of Data and Computer Science, Sun Yat-sen University, Guangzhou, China.

E-mail addresses: guangyuzhg@foxmail.com (G.-Y. Zhang), zhouyuren@mail.sysu.edu.cn (Y.-R. Zhou), hxyokokok@foxmail.com (X.-Y. He), changdongwang@hotmail.com (C.-D. Wang), huangdonghere@gmail.com (D. Huang).

The recently proposed subspace-based methods tend to exploit the underlying structure among different views with self-expression representation. The previous subspace-based methods usually follow a two-step strategy, i.e., to obtain a coefficient matrix in each view and then merge them to a common affinity matrix of all views. Representative methods include Diversity-induced Multi-view Subspace Clustering (DiMSC) [20] and Low-rank Tensor constrained Multiview Subspace Clustering (LT-MSC) [21]. Their limitations mainly lie in three aspects. First, the two-step strategy based methods fail to learn the optimal common affinity matrix across multiple views. Second, the negative coefficients cannot be avoided in the optimization process. Therefore, the existing methods usually force the negative coefficients to be positive, which may damage the inherent correlations in the data samples. What is more, the existing subspace-based methods consider the clustering problem only in the multiple linear subspace and they fail to handle the multi-view data with nonlinear manifolds.

To address above problems, in this paper, we propose a new multi-view subspace clustering method, named One-step Kernel Multi-view Subspace Clustering (OKMSC). Unlike the most existing subspace methods which adopt a two-step strategy, the proposed method jointly learns the coefficient matrix in each view as well as the common affinity matrix of all views under a one-step framework. In this way, these two matrix learning processes can be beneficial to each other, such that an optimal common affinity matrix can be obtained for achieving robust clustering results. In summary, the proposed method has the following advantages over the other methods.

- We introduce the probabilistic simplex on two variables, i.e., the coefficient vector for each view and the column vector of common affinity matrix, each of which is kept in the same scale. The probabilistic simplex constraint can keep properties of probability in these two variables, while the nonnegative constraint maintains the inherent correlation among the data samples, since the coefficient matrix and the common affinity matrix are naturally nonnegative.
- To explore the nonlinear relationship in the data, the proposed method transforms the data from original space to a kernel space, which improves the robustness of multi-view subspace clustering when faced with complex (especially nonlinear) data distributions.
- In order to learn a common graph of all views with exactly k connected components (where k is the number of clusters), we explicitly impose a rank constraint on the common affinity matrix in the optimization model. This design makes the data structure can be easily detected by a spectral clustering.
- We reformulate the proposed method as a constrained optimization problem, and solve this problem with Alternating Direction Method of Multipliers (ADMM) algorithm. By this way, each variable is updated efficiently and the convergence of the algorithm can be guaranteed. Furthermore, we conduct extensive experiments on several data sets, and the clustering results demonstrate the superiority of proposed method over the existing popular clustering methods.

The remainder of the paper is organized as follows. To begin with, we briefly review the background of multi-view clustering in Section 2. Then, in Section 3, we introduce the proposed OKMSC method with an ADMM based solution. In Section 4, we conduct several experiments on real-world multi-view data sets, and report the experimental results of OKMSC as well as several state-of-the-art methods. Finally, the conclusion is given in Section 5.

2. Related work

In the past decades, several efforts have dedicated to solve the multi-view clustering problem. In general, compared to single view setting, multiple views can provide more complementary information, which turns out to be beneficial to the clustering tasks. To exploit the complementary information of multiple views, early works focus on the two-view cases. Bickel et al. [22] extended k -means to cluster text data with two conditionally independent views. To the best of our knowledge, Bickel is the first to investigate the problem of multi-view clustering. De et al. [23] proposed a simple but effective spectral clustering method that can handle the two-view cases in the domain of web page. This method first utilizes the similarity matrix to combine two types of features, and then uses the spectral clustering to achieve the clustering result. To further exploit the complementary information of multiple views, a variety of works have been developed to handle the cases with more than two views. For example, Cai et al. [12] proposed a consistency based clustering method termed Robust Multi-view k -means Clustering (RMKC), which extends k -means by learning a common cluster assignment matrix across multiple views. This method is applicable to large scale clustering problem that can be easily parallelized. In [24], Tzortzis and Likas developed a kernel-based multi-view clustering, called Multi-View Kernel k -means (MVKKM). This method expresses multiple modalities through given kernel matrices, and all kernels are learned to clustering with a weighted combination strategy. Xu et al. [25] presented a novel weighted clustering method that can perform multi-view clustering and feature selection simultaneously. Inspired by max-product belief propagation, Wang et al. [26] proposed a novel multi-view clustering method termed Multi-View Affinity Propagation (MVAP). Based on the assumption that weighting strategy can improve the performance of multi-view clustering methods, Zhang et al. [27] developed a multi-view clustering method termed Two-level Weighted Collaborative k -means (TW-Co- k -means) by considering the consistency across multiple views to achieve clustering result. Huang et al. [28] further proposed a novel multi-view k -means clustering method, which utilizes the capped-norm loss to remove the data outliers and noise. Based on the collaborative strategy, Jiang et al. [29] presented a novel multi-view fuzzy clustering method, which incorporates the mutual links among different views and fuzzy clustering from each view into a joint optimization framework. Besides, inspired by the idea of co-clustering [30], Qian et al. [31] developed a collaborative multi-view clustering method based on inter-view collaborative learning and intra-view weighted attributes. To address the view-insufficiency issue in multi-view clustering, Huang et al. [32] presented a novel clustering method termed multi-view intact space clustering, which is able to jointly exploit the latent intact space from multiple insufficient views and discover the cluster structure of the latent space by matrix factorization. Recently, Zhang et al. [33] proposed a novel Binary Multi-View Clustering (BMVC) method which can easily scale to large-scale multi-view data. In BMVC, two key components, i.e., compact collaborative discrete representation learning and binary clustering structure learning are integrated into a unified learning framework. Inspired by the success of multi-task multi-view learning [34], Zhang et al. [35] further proposed a novel clustering method for multi-task multi-view clustering, which takes the advantages of Locally Linear Embedding (LLE) [36] and Laplacian Eigenmaps (LE) [37] to learn the structures in views of each task and different tasks, respectively. The aforementioned methods adapted different strategies to incorporate multiple views, and their effective have been proved in the different practical applications. Due to the well-motivated mathematical

framework, spectral-based methods (some papers refer to graph-based methods) have been widely investigated in the literature of multi-view clustering. One typical multi-view spectral clustering method was developed by Kumar and Daume [38]. This method is based on the co-training strategy which learns the graph structure in one view by using the clustering information from the other views. Kumar et al. [15] further proposed another novel multi-view spectral clustering method by enforcing the eigenvectors from all views towards a common consensus. Zhan et al. [39] developed a graph-based multi-view spectral clustering method termed Multi-view Clustering with Graph Learning (MCGL). The main idea is to learn a view-specific graph in each view first, and then fusing these input graphs to produce a global graph across as final result. However, the common shortcomings of the aforementioned methods can be concluded from two aspects. On the one hand, they usually construct the graph of each view individually, and hence the complementary information across multiple views is totally ignored. On the other hand, their performance is highly dependent on the quality of the constructed graph in each view.

Apart from the spectral-based methods, another popular research direction for multi-view clustering lies in the subspace-based methods. In [17], Gao et al. extended subspace clustering to multi-view cases, in which the common cluster structure is learned to enhance the consistency across multiple views. In [20], Cao et al. developed a novel subspace-based clustering method termed Diversity-induced Multi-view Subspace Clustering (DiMSC). The key idea is to explore the complementary information among different views with Hilbert Schmidt Independence Criterion (HSIC). To reveal the hidden structures of multi-view data, Huang et al. [40] proposed an auto-weighted subspace clustering method, which can simultaneously perform multi-view clustering task and learn the common graph across different views. Recently, Zhang et al. [41] developed a novel multi-view subspace clustering method, which seeks to learn the latent comprehensive representation while exploring the complementarity encoded in multiple views simultaneously. Although existing multi-view subspace clustering methods have been well investigated, they still have three limitations. First, for the most existing subspace-based methods, the magnitudes of subspace representations from different views are not in the same scope due to the lack of simplex constraints. Second, current methods usually adapt a two-step strategy to output common affinity matrix, which cannot guarantee that the obtained common affinity matrix is optimal. Third, these methods only consider the consistency or the diversity property among different views. This may lead to an unsatisfactory clustering result since merely considering consistency or diversity is not adequate in real world applications.

3. Proposed method

3.1. Notations

Before proposing our method, let us introduce some additional notations. All the matrices are written in capital letters. For a matrix M , its i th row, j th column, and ij -th element are denoted as $M_{i:}$, $M_{:j}$ and M_{ij} , respectively. The trace of matrix M is defined as $\text{Tr}(M)$. The transpose and the inverse of matrix M are denoted as M^T and M^{-1} , respectively. $\mathbf{1}_n$ is denoted as a n -dimensional column vector where each element is 1 and the letter I represents the identity matrix.

3.2. Subspace clustering

In this subsection, we briefly review some subspace clustering methods, which are mostly related to the proposed method.

Given a data set $X = \{x_1, \dots, x_n\} \in \mathbb{R}^{d \times n}$, where n represents the number of data samples and d represents the dimensional of feature space, respectively. The key idea of subspace clustering is the self-expressive property, which means that each data sample x_i can be expressed by a linear combination of the other samples. That is, x_i can be represented as follows:

$$x_i = \sum_{j \neq i} x_j z_{ji}, \quad (1)$$

where $Z = \{z_1, \dots, z_n\} \in \mathbb{R}^{n \times n}$ is the representation coefficient matrix and each column z_i is the coefficient vector corresponding to data sample x_i . To ensure that the representation coefficient matrix is desirable, several matrix norms on Z have been introduced to regularize subspace clustering, such as the l_1 norm [42], the Frobenius norm [43], and the nuclear norm [44]. The general optimization models of subspace clustering can be formulated as follows:

$$\min_Z \|Z\|_p \quad \text{s.t. } X = XZ, (\text{diag}(Z) = 0), \quad (2)$$

where $\|\cdot\|_p$ is the matrix norm constraint and $\text{diag}(Z)$ means that the diagonal elements of Z are enforced to zero. After obtaining the coefficient matrix, we can compute the affinity matrix as follows:

$$S = (|Z| + |Z^T|)/2. \quad (3)$$

Once the affinity matrix S is obtained, spectral clustering is performed on it to obtain the final clustering results.

3.3. Twin Learning for Similarity and Clustering (TLSC)

Our method is an adaption of TLSC that scales to multi-view framework. Therefore, in this subsection, we give a review of TLSC before introducing our framework. Different from the subspace clustering framework in Eq. (2), TLSC adopts a one-step framework which is able to learn the graph similarity and the cluster indicator simultaneously. In this way, it seamlessly incorporates two tasks, i.e., graph learning and cluster indicator matrix construction into a joint framework, where one task can be boosted by using the result of the other task. Moreover, to better extract the nonlinear relationship inherent in real-world data, TLSC extends the original linear space to kernel space by using single and multiple kernel learning methods. Concretely, given a data set $X = [x_1, \dots, x_n]$ and its kernel matrix $K_{ij} = \langle \phi(x_i), \phi(x_j) \rangle$, the objective function of TLSC is formulated as follows:

$$\min_{Z, F} \sum_{v=1}^V \|\phi(X) - \phi(X)Z\|_F^2 + \alpha \sum_{v=1}^V \|Z\|_F^2 + \beta \text{Tr}(F^T L F) \quad (4)$$

$$\text{s.t. } Z^T \mathbf{1}_n = \mathbf{1}_n, Z \geq 0, \text{diag}(Z) = 0, \forall v, F^T F = I,$$

where α and β are two balanced parameters controlled by users. Z is the representation matrix and F is the cluster indicator matrix. $L = D - Z$ is the Laplacian matrix, where D is the diagonal matrix with elements $D_{ii} = \sum_{j=1}^n S_{ij}$.

TLSC has obtained promising clustering results in face recognition tasks and its variants have been successfully used in various applications. However, it is developed for single-view clustering and is incapable of capturing the complementary information from multiple views.

3.4. The objective function

In this subsection, we introduce the proposed multi-view subspace clustering model. For multi-view data set, denote $X = [X^{(1)T}, \dots, X^{(V)T}]^T$ as the data matrix with V views. $X^{(v)} = [x_1^{(v)}, \dots, x_n^{(v)}] \in \mathbb{R}^{d_v \times n}$ is the v th view data matrix with d_v features. Assume there exists a nonlinear function $\phi(\cdot)$ that maps the data samples from original space to kernel space. Then the v th view data matrix $X^{(v)}$ can be transformed to $\phi(X^{(v)}) = [\phi(x_1^{(v)}), \dots, \phi(x_n^{(v)})]$. For the v th view, we denote $K^{(v)} \in \mathbb{R}^{n \times n}$ as the kernel matrix, where its elements represent the kernel similarity between data samples in the v th view. The v th kernel matrix $K^{(v)}$ can be computed as:

$$K_{ij}^{(v)} = \langle \phi(x_i^{(v)}), \phi(x_j^{(v)}) \rangle. \quad (5)$$

After computing the kernel matrix in each view, we can easily extend single view subspace clustering to multiple view domain. The objective function of kernel multi-view subspace clustering can be formulated as follows:

$$\min_{Z^{(v)}, E^{(v)}} \sum_{v=1}^V (\|E^{(v)}\|_F^2 + \alpha \|Z^{(v)}\|_F^2) \quad (6)$$

$$\text{s.t. } \phi(X^{(v)}) = \phi(X^{(v)})Z^{(v)} + E^{(v)}, \text{diag}(Z^{(v)}) = 0, \forall v,$$

where $Z^{(v)} = [z_1^{(v)}, \dots, z_n^{(v)}]$ is the representation matrix in the v th view and $z_i^{(v)}$ is the coefficient vector corresponding to the i th data sample in the v th view. $E^{(v)}$ represents the reconstruction error in the v th view. Parameter α is a balanced parameter. Constraint $\text{diag}(Z^{(v)}) = 0$ forces the diagonal elements of $Z^{(v)}$ to be zero, which means that each data sample cannot be linearly represented by itself. The model in Eq. (6) can be regarded as a nonlinear vision of Least Squares Regression (LSR) [43] in multi-view setting, which performs subspace clustering in individual view separately. After computing the representation matrix $Z^{(v)}$ for each view, we can obtain the common affinity matrix across multiple views as follows:

$$S = \sum_{v=1}^V (|Z^{(v)}| + |Z^{(v)T}|)/2. \quad (7)$$

However, in such a method, the consistency across different views is totally neglected. This leads to the loss of complement between different views. Besides, this method forces the non-negative elements in $Z^{(v)}$ to be positive by absolute operation, and this operation may damage the inherent correlation among data samples. Furthermore, the magnitude of elements in $Z^{(v)}$ may be significantly different due to the diversity of multiple views. Therefore, directly adding up the representation matrix $Z^{(v)}$ for individual view may output a poor common affinity S . To achieve a promising common affinity across multiple views, it is better to consider both the intra-view information and the consistency information simultaneously. Besides, the magnitude of elements in $Z^{(v)}$ should be in the same scope. Inspired by the recent research in [45], we propose a unified model to address above problems, which aims to explore the diversity and consistency across different views with simplex representation. The new objective function can be written as:

$$\min_{Z^{(v)}, S} \sum_{v=1}^V \|E^{(v)}\|_F^2 + \alpha \sum_{v=1}^V \|Z^{(v)}\|_F^2 + \beta \sum_{v=1}^V \|S - Z^{(v)}\|_F^2 \quad (8)$$

$$\text{s.t. } \phi(X^{(v)}) = \phi(X^{(v)})Z^{(v)} + E^{(v)}, \text{diag}(Z^{(v)}) = 0, \forall v,$$

$$Z^{(v)T} \mathbf{1}_n = \mathbf{1}_n, Z^{(v)} \geq 0, \forall v, S^T \mathbf{1}_n = \mathbf{1}_n, S \geq 0.$$

Here $S = [s_1, \dots, s_n]$ denotes the common affinity matrix and α, β are two balancing parameters. We use the simplex

representation here since the constraints on the column vectors of $Z^{(v)}$ and S are simplex, i.e., nonnegative and sum up to one. By this way, the magnitudes of elements in $Z^{(v)}$ and S are in the same scope, and $Z^{(v)}$ can measure the similarity of data samples via learning the global structure of data in the v th view.

Furthermore, we expect the common affinity matrix S to have exactly k connect components, where k is the cluster number of the given data set. To achieve this goal, we can impose a rank constraint on the Laplacian matrix of S . According to the Theorem in [46], if Laplacian matrix L satisfies the constraint $\text{rank}(L) = n - k$, then affinity matrix S contains exactly k connected components. Since Laplacian matrix L is positive semidefinite, then its i th smallest eigenvalue $\sigma_i \geq 0$. According to the research in [9], constraint $\text{rank}(L) = n - k$ is equivalent to $\sum_{i=1}^k \sigma_i(L) = 0$. Then, the new objective function is computed via:

$$\min_{Z^{(v)}, S} \sum_{v=1}^V \{\|E^{(v)}\|_F^2 + \alpha \|Z^{(v)}\|_F^2 + \beta \|S - Z^{(v)}\|_F^2\} + \gamma \sum_{i=1}^k \sigma_i \quad (9)$$

$$\text{s.t. } \phi(X^{(v)}) = \phi(X^{(v)})Z^{(v)} + E^{(v)}, \text{diag}(Z^{(v)}) = 0, \forall v,$$

$$Z^{(v)T} \mathbf{1}_n = \mathbf{1}_n, Z^{(v)} \geq 0, \forall v, S^T \mathbf{1}_n = \mathbf{1}_n, S \geq 0.$$

Furthermore, following the Ky Fan's theorem [47], we have the equation:

$$\sum_{i=1}^k \sigma_i = \min_{F^T F = I} \text{Tr}(F^T L F), \quad (10)$$

where $F \in \mathbb{R}^{n \times k}$ is the cluster indicator matrix, and its elements indicate the membership of data samples belonging to the k clusters. The Laplacian matrix $L = D - (S + S^T)/2$, where the diagonal matrix D represents the degree matrix and its i th element is $\sum_{j=1}^n (S_{ij} + S_{ji})/2$. Finally, by utilizing Eqs. (9) and (10), the objective function of proposed method OKMSC can be written as follows:

$$\min_{Z^{(v)}, S, F} \underbrace{\sum_{v=1}^V \|\phi(X^{(v)}) - \phi(X^{(v)})Z^{(v)}\|_F^2 + \alpha \sum_{v=1}^V \|Z^{(v)}\|_F^2}_{\text{Intra-view structure learning}} \quad (11)$$

$$+ \underbrace{\beta \sum_{v=1}^V \|S - Z^{(v)}\|_F^2 + \gamma \text{Tr}(F^T L F)}_{\text{Inter-view consistency learning}}$$

$$\text{s.t. } Z^{(v)T} \mathbf{1}_n = \mathbf{1}_n, Z^{(v)} \geq 0, \text{diag}(Z^{(v)}) = 0, \forall v,$$

$$S^T \mathbf{1}_n = \mathbf{1}_n, S \geq 0, F^T F = I.$$

The proposed optimization model consists of two parts. The first part is the intra-view structure learning, which aims to learn the subspace structure in each view. The second part is the inter-view consistency learning, which measures the correlation across different views. By exploring both the view-specific property and view-consistency across multi-view data, our unified model can learn both the intra-view subspace structure and common cluster structure simultaneously. In this way, the proposed method can achieve the optimal common affinity matrix across multiple views that products promising clustering results. Fig. 1 shows the main framework of proposed method.

3.5. Optimization

In order to apply ADMM [48,49] to solve the optimization problem in Eq. (11), we introduce V auxiliary variables $C^{(v)}$, $v =$

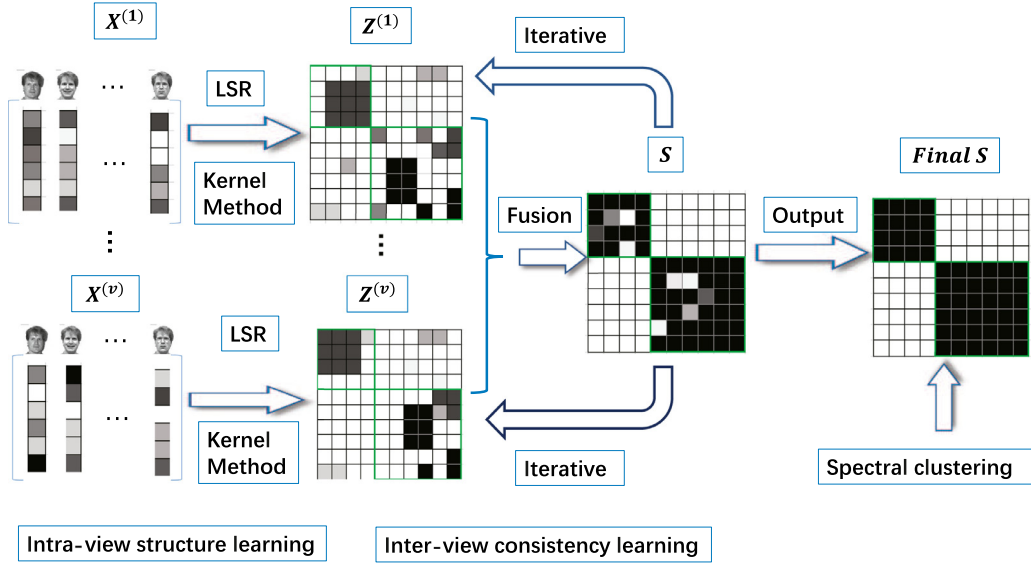


Fig. 1. The framework of proposed method.

1, ..., V to the OKMSC model. By combining the kernel function in Eq. (5), Eq. (11) is equivalently to the following problem:

$$\begin{aligned} \min_{Z^{(v)}, S, C^{(v)}, F} \sum_{v=1}^V & \text{Tr}(K^{(v)} - 2K^{(v)}Z^{(v)} + Z^{(v)T}K^{(v)}Z^{(v)}) \\ & + \alpha \sum_{v=1}^V \|C^{(v)}\|_F^2 + \beta \sum_{v=1}^V \|S - C^{(v)}\|_F^2 + \gamma \text{Tr}(F^T L F) \\ \text{s.t. } & C^{(v)T} \mathbf{1}_n = \mathbf{1}_n, C^{(v)} \geq 0, C^{(v)} = Z^{(v)}, \text{diag}(Z^{(v)}) = 0, \forall v, \\ & S^T \mathbf{1}_n = \mathbf{1}_n, S \geq 0, F^T F = I. \end{aligned} \quad (12)$$

It can be solved by alternating direction method of multipliers (ADMM). The corresponding augmented Lagrangian function is as follows:

$$\begin{aligned} \mathcal{L}(Z^{(v)}, C^{(v)}, Y^{(v)}, S, F) \\ = \sum_{v=1}^V & \text{Tr}(-2K^{(v)}Z^{(v)} + Z^{(v)T}K^{(v)}Z^{(v)}) + \alpha \sum_{v=1}^V \|C^{(v)}\|_F^2 + \\ & \beta \sum_{v=1}^V \|S - C^{(v)}\|_F^2 + \sum_{v=1}^V \langle Y^{(v)}, C^{(v)} - Z^{(v)} \rangle + \\ & \frac{\mu}{2} \sum_{v=1}^V \|C^{(v)} - Z^{(v)}\|_F^2 + \gamma \text{Tr}(F^T L F), \end{aligned} \quad (13)$$

where $Y^{(v)}$ is the Lagrangian multiplier for the v th view and $\mu \geq 0$ is a plenty parameter. We can alternatively update one variable by keeping the others fixed in an iteration.

3.5.1. Update $Z^{(v)}$ while keeping other variables

We can update variable $Z^{(v)}$ by solving the following problem:

$$\begin{aligned} \min_{Z^{(v)}} & \text{Tr}(-2K^{(v)}Z^{(v)} + Z^{(v)T}K^{(v)}Z^{(v)}) + \\ & \frac{\mu}{2} \|Z^{(v)} - (C^{(v)} + \frac{1}{\mu}Y^{(v)})\|_F^2 \quad \text{s.t. } \text{diag}(Z^{(v)}) = 0. \end{aligned} \quad (14)$$

The closed solution for $Z^{(v)}$ is given as:

$$Z^{(v)} = \tilde{Z}^{(v)} - \text{diag}(\tilde{Z}^{(v)}), \quad (15)$$

Algorithm 1 Projection w onto the probabilistic simplex

Input: A vector w .

- 1: Sort w into v : $v_1 \geq v_2 \geq \dots \geq v_p$.
- 2: Find $\theta = \max\{1 \leq j \leq p : v_j + \frac{1}{j}(1 - \sum_{i=1}^j v_i) > 0\}$.
- 3: Define $\lambda = \frac{1}{\theta}(1 - \sum_{i=1}^{\theta} v_i)$.
- 4: Compute $u = \max\{w + \lambda, 0\}$.

Output: A vector u .

where the (i, j) entry of $\tilde{Z}^{(v)}$ is given by:

$$\tilde{Z}^{(v)} = (K^{(v)} + \frac{\mu}{2}I)^{-1}(K^{(v)} + \frac{\mu}{2}C^{(v)} + \frac{1}{2}Y^{(v)}). \quad (16)$$

3.5.2. Update $C^{(v)}$ while keeping other variables

We can update variable $C^{(v)}$ by solving the following problem:

$$\begin{aligned} \min_{C^{(v)}} & \|C^{(v)} - \frac{1}{2\alpha + 2\beta + \mu}(2\beta S + \mu Z^{(v)} - Y^{(v)})\|_F^2 \\ \text{s.t. } & C^{(v)T} \mathbf{1}_n = \mathbf{1}_n, C^{(v)} \geq 0. \end{aligned} \quad (17)$$

This is a quadratic problem and it can be solved by projection based method. In this way, the solution of the i th column $C_{:,i}^{(v)}$ of variable $C^{(v)}$ can be solved by projecting it into a probabilistic simplex. The solution of problem (17) can be solved by an efficient iterative Algorithm 1 [50].

3.5.3. Update S while keeping other variables

We can update variable S by solving the following problem:

$$\begin{aligned} \min_S & \|S - \frac{1}{V}(\sum_{v=1}^V C^{(v)} - \frac{\gamma}{2\beta}H)\|_F^2 \\ \text{s.t. } & S^T \mathbf{1}_n = \mathbf{1}_n, S \geq 0. \end{aligned} \quad (18)$$

Here H is an auxiliary matrix and its element is defined as $H_{ij} = \|F_{:,i} - F_{:,j}\|_2^2$. Similar to the optimization of problem (17), we can obtain the unique solution of above problem by Algorithm 1.

3.5.4. Update F while keeping other variables

We can update variable F by solving the following problem:

$$\begin{aligned} \min_{F^T F = I} & \text{Tr}(F^T L F). \end{aligned} \quad (19)$$

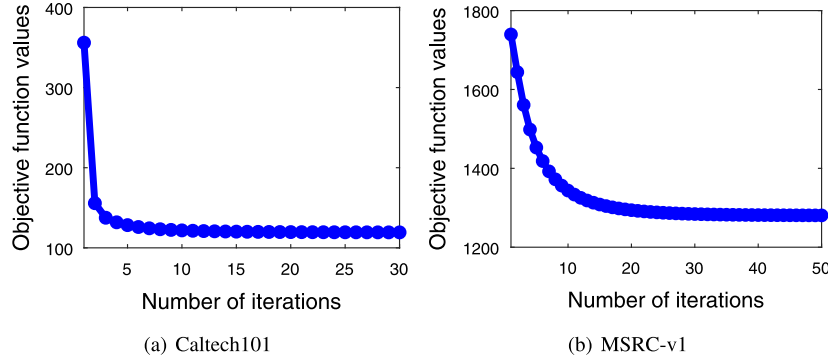


Fig. 2. Convergence curves of OKMSC on two benchmark data sets.

Algorithm 2 OKMSC

Input: Data Set $X = \{X^{(1)}, \dots, X^{(V)}\}$ with V views, the number of clusters k , the maximum number of iterations t_{max} .

Parameter: Balancing parameters $\alpha, \beta, \gamma > 0$ and plenty parameter $\mu = 1.2$ or 1.1 .

- 1: **Initialization:** Set $t = 1$. Initialize $Z^{(v)} = I, \forall v, Y^{(v)} = I, \forall v$ and $S = I$. Initialize F such that $F^T F = I$.
- 2: **repeat**
- 3: Update $Z^{(v)}$ in the t -th view by Eqs. (15) (16).
- 4: Update $C^{(v)}$ in the t -th view by solving problem (17).
- 5: Update S by solving problem (18).
- 6: Update F by solving problem (19).
- 7: Update $Y^{(v)}$ in the t -th view by Eq. (20).
- 8: $t = t + 1$.
- 9: **until** Convergence or $t > t_{max}$
- 10: Perform spectral clustering on the common affinity matrix S to obtain the final clustering result.

Output: Final clustering result, $Z^{(v)}, \forall v, S$ and F .

The optimal solution of F can be obtained by the k eigenvectors of L corresponding to the k smallest values.

3.5.5. Update $Y^{(v)}$ while keeping other variables

We can update variable $Y^{(v)}$ by using the following equation:

$$Y^{(v)} = Y^{(v)} + \mu(C^{(v)} - Z^{(v)}). \quad (20)$$

We repeat above five steps until the objective function in Eq. (11) converges to local minimum or the number of iterations reaches the predefined threshold t_{max} . The detail optimization procedure is described in Algorithm 2.

3.6. Time complexity and convergence

First, we analyze the time complexity of our method by the big O notation. There are five variables, i.e., $Z^{(v)}, C^{(v)}, S, F$ and $Y^{(v)}$, in our method. In each iteration, the costs for updating $Z^{(v)}, C^{(v)}, S, F$ and $Y^{(v)}$ are $O(n^3), O(n^2 \log(n)), O(n^2 \log(n)), O(kn^2)$ and $O(1)$, respectively (where n represents the number of samples and k represents the number of clusters). In the step of updating $Z^{(v)}$, we only need to pre-calculate the inverse of matrix $(K^{(v)} + \frac{\mu}{2}I)$ once before iteration starts. Assume there are V views in the data set and Algorithm 2 runs T times until it stops. In sum, as $V \ll n, V \ll T, k \ll n$ and $k \ll T$, the overall time complexity of our method is $O(Vn^3 + VTn^2 \log(n) + Tn^2 \log(n) + Tkn^2 + TV) = O(n^3 + Tn^2 \log(n))$. Based on the above analysis, it can be observed that the complexity of our method is mainly determined by the

size of data sample n . Inspired by the recent studies on large-scale clustering [51], we find one potential solution to reduce the complexity of our method, termed as hybrid representative selection strategy. This selection strategy has been successfully introduced into the fields like spectral clustering and ensemble clustering. In the future, we would like to introduce this strategy or other landmark-based strategy [52,53] into our model, aiming to promote the scalability and robustness of OKMSC for large-scale multi-view clustering problems.

Next, we study the convergence property of our method. To the best of our knowledge, it is still difficult to prove the strong convergence of ADMM optimization problem with unknown variables (especially more than three variables) [54,55]. Therefore, giving a strict theoretical analysis to prove the convergence of Algorithm 2 is difficult. However, some recently proposed works [55,56] have experimentally validated the convergence of ADMM framework with many variables. Following these works, we also study the convergence behavior of our method from the aspect of experiments. The convergence curves of our method on two benchmark data sets are shown in Fig. 2. It is observed that our method has relatively fast convergence speed and tends to be stable after 30 or 50 iterations. This implies relatively good convergence property of our method.

4. Experiments

In this section, we conduct extensive experiments to validate the effectiveness of proposed method from the aspects of comparison experiment, parameter analysis and complexity analysis. All the experiments are implemented on a PC with an Intel 3.4-GHz CPU and 64-GB RAM.

4.1. Experimental setting

In our experiments, we evaluate the effectiveness of proposed method OKMSC on six widely used multi-view image, face or document data sets, namely, Caltech101-7 [57], MSRC-v1 [57], Reuters [13], Yale [58], Cora [26] and VIS/NIR [26]. Figs. 3 to 4 show some example images from Yale and VIS/NIR data sets, respectively. The summarizations of these data sets are listed in Table 1 and the detailed descriptions are as follows.

- **Caltech101-7** is an object recognition data set that contains 8677 color images of 101 categories. In this paper, we select seven widely used categories, that is, Dolla-Bill, Face, Garfield, Motorbikes, Snoopy, Stop-Sign and Windsor-Chair. Meanwhile, we extract six visual features from each image, i.e., LBP, PHOG, GIST, Gabor, SURF and SIFT.

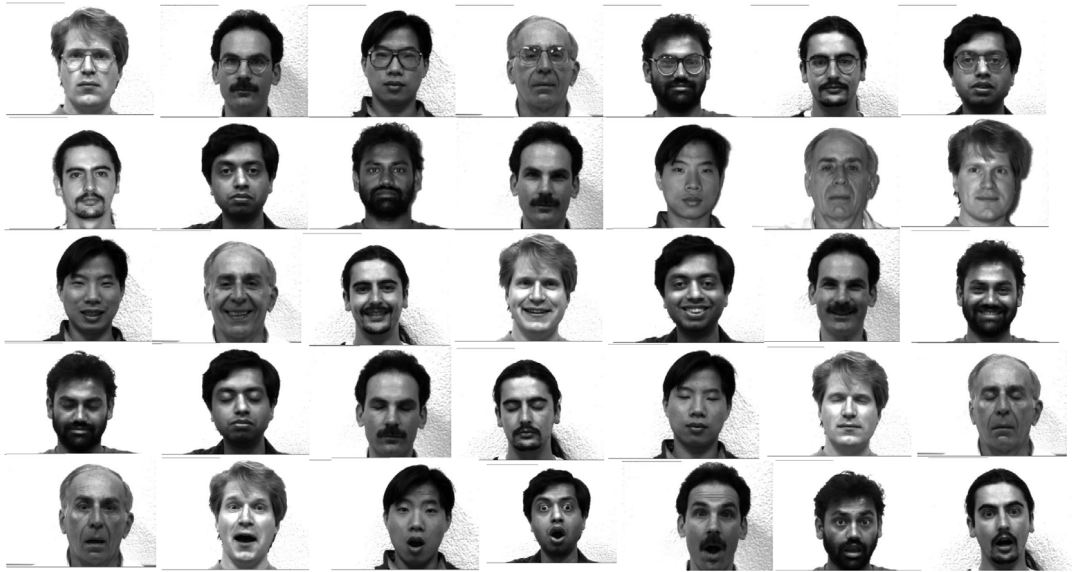


Fig. 3. Sample images of Yale.



Fig. 4. Sample images of VIS/NIR.

Table 1

The summarization of six benchmark data sets.

View type	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
1	LBP(256)	LBP(256)	English(9749)	Intensity(4096)	Cont(3000)	View 1(10000)
2	PHOG(680)	HOG(100)	French(9109)	LBP(3304)	Cite(1840)	View 2(10000)
3	GIST(512)	GIST(512)	German(7774)	Gabor(4096)	–	–
4	Gabor(32)	CENTRIST(1302)	–	–	–	–
5	SURF(200)	CMT(48)	–	–	–	–
6	SIFT(200)	SIFT(200)	–	–	–	–
Samples	441	210	600	165	1051	1056
Classes	7	7	6	15	2	22

- **MSRC101-v1** is a widely used data set for scene recognition, which contains total 8 classes and 240 images. In our experiments, we select seven classes including airplane, building, bicycle, cow, car, face and tree. Each image is described by six types of features, namely, LBP, HOG, GIST, CMT, CENTRIST and SIFT with visual features from each image.
- **Reuters** is a data set consists of multilingual documents belonging to six large Reuters categories. In our experiments, we use a randomly sampled subset containing 600 documents, with each category containing 100 documents. This data set contains three views, i.e., English, French and German.
- **Yale** is one of the most popular facial image data sets for multi-view clustering. This data set contains 165 gray images covered 15 classes, and each class has 11 images. The used classes include center-light, with glasses, happy, left-light, without glasses, normal, right-light, sad, sleepy, surprised and wink. In this paper, we extract three types of features from each image, i.e., intensity, LBP and Gabor.
- **Cora** is a link-based document data set contains two types of views, i.e., the document contents and document citations. It contains totally of 1051 machine learning papers, belonging to two classes.
- **VIS/NIR** is a multi-view facial data set that consists of 1056 images over 22 classes. It consists two types of views, where each view only captures partial information to describe the facial images.

In our experiments, we report the clustering results by using three evaluation metrics, i.e., Accuracy (Acc) [59], Normalized Mutual Information (NMI) [51] and F-score [60].

Given a data set X of size N , we set μ to be the clustering labels of K clusters and ν to be the true class labels of L classes. The NMI score of μ with respect to ν can be defined as follows:

$$\text{NMI}(\mu, \nu) = \frac{2 \sum_{k=1}^K \sum_{l=1}^L \frac{n_{kl}^2}{N} \log \frac{n_{kl}^2 N}{\sum_{i=1}^K n_{il}^2 \sum_{j=1}^L n_{kj}^2}}{H(\mu) + H(\nu)}, \quad (21)$$

where $H(\mu)$ and $H(\nu)$ are the Shannon entropy of cluster labels μ and class labels ν respectively, with n_i and n^j denoting the number of data points in cluster i and class j , and n_{ij}^2 represents the number of instances in cluster i and class j .

Acc is a popular evaluation metric which measures one-to-one relationship between clustering labels and the true class labels. Given a data set X of size N . Let μ be the clustering labels and ν be the true class labels. Then the definition of Acc can be defined as follows:

$$\text{Acc}(\mu, \nu) = \frac{\sum_{i=1}^N \delta(\mu, \text{map}(\nu))}{N}, \quad (22)$$

where $\delta(\cdot)$ is the delta function, and $\text{map}(\cdot)$ is the best permutation mapping function that projects each cluster label to the true label.

We need to obtain the precision and recall before computing the value of F-score. The formulations of the precision and recall are as follows:

$$\text{Precision}(C^\alpha, C^\beta) = \frac{N_{\alpha, \beta}}{|C_\alpha|}, \quad (23)$$

$$\text{Recall}(C^\alpha, C^\beta) = \frac{N_{\alpha, \beta}}{|C_\beta|}, \quad (24)$$

where C_α and C_β are two different clusters, $N_{\alpha, \beta}$ denotes the number of instances from C_α in the class C_β . $|C_\alpha|$ and $|C_\beta|$ represent the module of the class C_α and C_β respectively.

Table 2

Parameter setting of OKMSC on the six multi-view data sets.

	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
α	1	1	1	0.5	0.5	1.5
β	0.5	0.2	0.5	1	0.1	1
γ	0.05	0.001	0.001	0.01	0.05	0.01

Table 3

Kernel constructions of OKMSC on the six multi-view data sets.

	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
1	Gaussian	Linear	Linear	Gaussian	Linear	Linear
2	Linear	Linear	Linear	Linear	Linear	Linear
3	Gaussian	Linear	Gaussian	polynomial	–	–
4	Gaussian	Linear	–	–	–	–
5	Linear	polynomial	–	–	–	–
6	Linear	Gaussian	–	–	–	–

After computing the values of precision and recall, we can calculate the F-score between C_α and C_β as follows:

$$F(C_\alpha, C_\beta) = \frac{2 \text{Precision}(C_\alpha, C_\beta) \text{Recall}(C_\alpha, C_\beta)}{\text{Precision}(C_\alpha, C_\beta) + \text{Recall}(C_\alpha, C_\beta)}. \quad (25)$$

These three metrics favor different properties of the clustering results. For all these metrics, higher values indicate the better clustering performance.

4.2. Comparison experiments

In this subsection, we compare the proposed method with several representative alternatives, including classic k -means, two single view subspace clustering methods and six multi-view clustering methods. For k -means, we use it as the baseline method. More concretely, we apply k -means on each view of multi-view data (e.g. KM(1) represents the clustering result by performing k -means on view 1) as well as the concatenated features of all views (e.g. KM(Allfea) represents the clustering result by performing k -means on the concatenated features of multi-view data). For the single-view methods, we select two recently developed algorithms for comparison, namely, Structured Sparse Subspace Clustering (S³C) [61] and Twin Learning for Similarity and Clustering (TLSC) [62]. In particular, TLSC is a kernel subspace clustering method with graph rank constraint, which is related to our method. These single-view methods first concatenate the features of all views and then perform the clustering algorithms. For the multi-view methods, six representative algorithms are used for comparison, namely, Multi-view Learning with Adaptive Neighbors (MLAN) [8], Self-weighted Multiview Clustering (SwMC) [19], Robust Multi-view Spectral Clustering (RMSC) [16], Diversity-induced Multi-view Subspace Clustering (DiMSC) [20], Graph-based System for Multi-view Clustering (GSC) [63] and Latent Multi-view Subspace Clustering (LMSC) [41,64]. Specially, for the two-step graph based methods, namely, SwMC, we follow Constrained Laplacian Rank (CLR) [65] to construct the affinity matrix for each view as the initial input matrix. Besides, for the two-step subspace method, namely, DiMSC, we compute its global affinity matrix via Eq. (7). The descriptions of these baseline clustering methods are as follows.

- **Structured Sparse Subspace Clustering (S³C)**: This method extends sparse subspace clustering (SSC) by incorporating affinity matrix learning and spectral clustering into a unified framework.
- **Twin Learning for Similarity and Clustering (TLSC)**: TLSC is a subspace clustering method which can perform spectral clustering tasks and learn the similarity relationship in kernel space simultaneously.

Table 4
Average NMI (%) of all methods on six multi-view data sets.

	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
KM (1)	0.323 $\pm$ 0.006	0.567 $\pm$ 0.015	0.295 $\pm$ 0.032	0.549 $\pm$ 0.028	0.112 $\pm$ 0.021	0.782 $\pm$ 0.009
KM (2)	0.464 $\pm$ 0.000	0.156 $\pm$ 0.000	0.231 $\pm$ 0.065	0.584 $\pm$ 0.039	0.176 $\pm$ 0.006	0.497 $\pm$ 0.013
KM (3)	0.660 $\pm$ 0.012	0.637 $\pm$ 0.017	0.273 $\pm$ 0.044	0.642 $\pm$ 0.023	–	–
KM (4)	0.579 $\pm$ 0.013	0.526 $\pm$ 0.010	–	–	–	–
KM (5)	0.270 $\pm$ 0.002	0.432 $\pm$ 0.000	–	–	–	–
KM (6)	0.428 $\pm$ 0.000	0.397 $\pm$ 0.015	–	–	–	–
KM (AllFea)	0.672 $\pm$ 0.007	0.785 $\pm$ 0.015	0.291 $\pm$ 0.032	0.623 $\pm$ 0.019	0.108 $\pm$ 0.000	0.583 $\pm$ 0.009
S ³ C	0.692 $\pm$ 0.002	0.753 $\pm$ 0.007	0.367 $\pm$ 0.000	0.656 $\pm$ 0.014	N/A	0.594 $\pm$ 0.014
TLSC	0.723 $\pm$ 0.024	0.792 $\pm$ 0.053	0.364 $\pm$ 0.017	0.584 $\pm$ 0.017	0.714 $\pm$ 0.007	0.642 $\pm$ 0.024
MLAN	0.704 $\pm$ 0.002	0.808 $\pm$ 0.000	0.260 $\pm$ 0.014	0.705 $\pm$ 0.007	0.125 $\pm$ 0.068	0.562 $\pm$ 0.005
SwMC	0.670 $\pm$ 0.000	0.836 $\pm$ 0.000	0.230 $\pm$ 0.002	0.656 $\pm$ 0.001	0.263 $\pm$ 0.000	0.961 $\pm$ 0.005
RMSC	0.689 $\pm$ 0.021	0.641 $\pm$ 0.044	0.344 $\pm$ 0.030	0.529 $\pm$ 0.028	0.685 $\pm$ 0.002	0.727 $\pm$ 0.037
DiMSC	0.745 $\pm$ 0.004	0.688 $\pm$ 0.007	0.304 $\pm$ 0.030	0.684 $\pm$ 0.031	N/A	0.868 $\pm$ 0.017
GSC	0.589 $\pm$ 0.000	0.893 $\pm$ 0.000	0.249 $\pm$ 0.000	0.689 $\pm$ 0.000	0.217 $\pm$ 0.000	0.985 $\pm$ 0.000
LMSC	0.754 $\pm$ 0.066	0.767 $\pm$ 0.048	0.308 $\pm$ 0.030	0.768 $\pm$ 0.019	0.724 $\pm$ 0.011	0.978 $\pm$ 0.039
OKMSC	0.786 $\pm$ 0.002	0.847 $\pm$ 0.000	0.396 $\pm$ 0.000	0.772 $\pm$ 0.009	0.740 $\pm$ 0.010	0.990 $\pm$ 0.002

Table 5
Average Acc (%) of all methods on six multi-view data sets.

	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
KM (1)	0.386 $\pm$ 0.004	0.714 $\pm$ 0.014	0.430 $\pm$ 0.037	0.520 $\pm$ 0.035	0.658 $\pm$ 0.040	0.680 $\pm$ 0.017
KM (2)	0.537 $\pm$ 0.001	0.295 $\pm$ 0.001	0.393 $\pm$ 0.062	0.550 $\pm$ 0.039	0.822 $\pm$ 0.002	0.276 $\pm$ 0.013
KM (3)	0.721 $\pm$ 0.004	0.729 $\pm$ 0.054	0.434 $\pm$ 0.042	0.588 $\pm$ 0.032	–	–
KM (4)	0.654 $\pm$ 0.007	0.608 $\pm$ 0.013	–	–	–	–
KM (5)	0.392 $\pm$ 0.001	0.538 $\pm$ 0.002	–	–	–	–
KM (6)	0.496 $\pm$ 0.001	0.496 $\pm$ 0.021	–	–	–	–
KM (AllFea)	0.731 $\pm$ 0.003	0.835 $\pm$ 0.012	0.449 $\pm$ 0.030	0.562 $\pm$ 0.027	0.782 $\pm$ 0.001	0.423 $\pm$ 0.013
S ³ C	0.675 $\pm$ 0.000	0.803 $\pm$ 0.003	0.510 $\pm$ 0.025	0.634 $\pm$ 0.022	N/A	0.431 $\pm$ 0.013
TLSC	0.747 $\pm$ 0.004	0.821 $\pm$ 0.090	0.511 $\pm$ 0.028	0.541 $\pm$ 0.023	0.945 $\pm$ 0.008	0.653 $\pm$ 0.016
MLAN	0.727 $\pm$ 0.007	0.838 $\pm$ 0.000	0.278 $\pm$ 0.015	0.676 $\pm$ 0.008	0.776 $\pm$ 0.013	0.367 $\pm$ 0.002
SwMC	0.694 $\pm$ 0.000	0.910 $\pm$ 0.000	0.258 $\pm$ 0.000	0.643 $\pm$ 0.002	0.851 $\pm$ 0.002	0.916 $\pm$ 0.009
RMSC	0.730 $\pm$ 0.033	0.702 $\pm$ 0.067	0.514 $\pm$ 0.032	0.434 $\pm$ 0.031	0.959 $\pm$ 0.000	0.597 $\pm$ 0.051
DiMSC	0.724 $\pm$ 0.004	0.757 $\pm$ 0.003	0.457 $\pm$ 0.003	0.658 $\pm$ 0.035	N/A	0.846 $\pm$ 0.031
GSC	0.617 $\pm$ 0.000	0.943 $\pm$ 0.000	0.303 $\pm$ 0.000	0.655 $\pm$ 0.000	0.806 $\pm$ 0.000	0.955 $\pm$ 0.000
LMSC	0.757 $\pm$ 0.061	0.813 $\pm$ 0.054	0.453 $\pm$ 0.050	0.742 $\pm$ 0.024	0.958 $\pm$ 0.024	0.943 $\pm$ 0.060
OKMSC	0.776 $\pm$ 0.004	0.919 $\pm$ 0.000	0.524 $\pm$ 0.000	0.757 $\pm$ 0.008	0.967 $\pm$ 0.011	0.981 $\pm$ 0.003

- **Multi-view Learning with Adaptive Neighbors (MLAN):** MLAN is an auto-weighted clustering method which simultaneously performs multi-view clustering task and local structure learning.
- **Self-weighted Multiview Clustering (SwMC):** SwMC is a popular multi-view clustering method that based on the parameter-free strategy. It aims to learn the global graph from a series of input graphs (each input graph corresponding to a view) with Laplacian rank constraint.
- **Robust Multi-view Spectral Clustering (RMSC):** RMSC is a Markov chain method for multi-view clustering, which aims to learn the global transition probability matrix across multiple views via low-rank and sparse decomposition.
- **Diversity-induced Multi-view Subspace Clustering (DiMSC):** It is a subspace clustering method for multi-view data, whose key idea is to explore the complementary information of different views with Hilbert Schmidt Independence Criterion.
- **Graph-based System for multi-view Clustering (GSC):** GSC is a graph-based multi-view clustering method that follows a two-step framework. It works by first constructing graph metrics from multiple views and then fusing them into a robust unified graph.
- **Latent Multi-view Subspace Clustering (LMSC):** LMSC is a novel and robust multi-view clustering method that aims to reveal cluster structure by jointly learning the latent representation and the subspace representation within a unified framework.

For all the compared methods, we download the source codes from the authors' websites, and tune the parameters in the same

way as suggested by the authors [8,16,19,20,61–64]. For the proposed method, we construct three types of kernels (each kernel corresponds to a view), including a linear kernel (i.e., $K(x, y) = x^T y$), two polynomial kernels (i.e., $K(x, y) = (a + x^T y)^b$ with $a = \{0, 1\}$ and $b = \{4\}$) and a Gaussian kernel (i.e., $K(x, y) = \exp(-\|x - y\|_2^2 / (d_{\max}^2))$, where $d_{\max}$ is the maximal distance between data samples). The corresponding experimental setups of proposed method include parameter settings, kernel constructions in different views, listed in Tables 2 and 3, respectively. For fairness, we repeat all the baseline clustering methods 30 times on each data set and report the average performance with the standard deviation. The best score in each metric is highlighted in bold.

Tables 4 to 6 report the clustering results of all methods on different multi-view data sets in terms of NMI, Acc and F-score, respectively. Specially, S³C and DiMSC are not applicable on the Cora data set. These tables enable us to observe following two findings. In the first place, the integration of non-linear kernel space and graph rank constraint can improve the clustering results of subspace clustering. This can be validated by comparing the performance of TLSC over the other multi-view clustering methods. Next, although the two-step multi-view methods SwMC, RMSC, DiMSC and GSC sometimes show competitive performance over the proposed unified method, these methods fail to achieve promising clustering on different multi-view data sets. What is more, the proposed method achieves the best clustering performance over the comparison methods on most of the six data sets. The reason is that the proposed method can learn an optimal affinity matrix across multiple views by simultaneously considering three key factors, i.e., nonlinear kernel

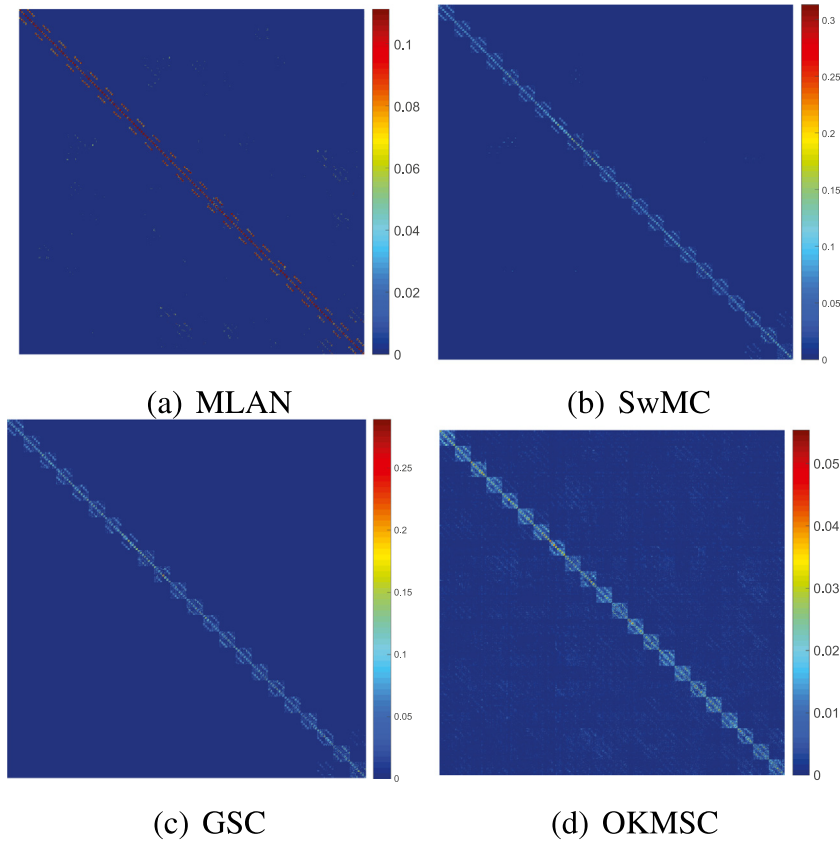


Fig. 5. An illustration of the common affinity matrix generated by different multi-view clustering methods on VIS/NIR data set.

Table 6

Average F-score (%) of all methods on six multi-view data sets.

	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
KM (1)	0.337 $\pm$ 0.002	0.530 $\pm$ 0.020	0.359 $\pm$ 0.019	0.360 $\pm$ 0.030	0.658 $\pm$ 0.041	0.365 $\pm$ 0.017
KM (2)	0.461 $\pm$ 0.000	0.194 $\pm$ 0.001	0.324 $\pm$ 0.037	0.390 $\pm$ 0.041	0.814 $\pm$ 0.001	0.213 $\pm$ 0.013
KM (3)	0.668 $\pm$ 0.010	0.618 $\pm$ 0.028	0.343 $\pm$ 0.025	0.456 $\pm$ 0.309	–	–
KM (4)	0.549 $\pm$ 0.007	0.494 $\pm$ 0.010	–	–	–	–
KM (5)	0.336 $\pm$ 0.001	0.404 $\pm$ 0.001	–	–	–	–
KM (6)	0.407 $\pm$ 0.000	0.361 $\pm$ 0.012	–	–	–	–
KM (AllFea)	0.684 $\pm$ 0.005	0.758 $\pm$ 0.018	0.357 $\pm$ 0.030	0.413 $\pm$ 0.025	0.794 $\pm$ 0.000	0.352 $\pm$ 0.012
S ³ C	0.535 $\pm$ 0.000	0.708 $\pm$ 0.023	0.406 $\pm$ 0.001	0.474 $\pm$ 0.021	N/A	0.371 $\pm$ 0.015
TLSC	0.718 $\pm$ 0.023	0.725 $\pm$ 0.003	0.389 $\pm$ 0.003	0.353 $\pm$ 0.039	0.936 $\pm$ 0.003	0.350 $\pm$ 0.028
MLAN	0.678 $\pm$ 0.003	0.729 $\pm$ 0.000	0.293 $\pm$ 0.005	0.521 $\pm$ 0.013	0.784 $\pm$ 0.002	0.277 $\pm$ 0.051
SwMC	0.610 $\pm$ 0.000	0.826 $\pm$ 0.000	0.289 $\pm$ 0.001	0.473 $\pm$ 0.002	0.807 $\pm$ 0.000	0.912 $\pm$ 0.051
RMSC	0.683 $\pm$ 0.036	0.612 $\pm$ 0.043	0.390 $\pm$ 0.030	0.336 $\pm$ 0.034	0.942 $\pm$ 0.000	0.519 $\pm$ 0.053
DiMSC	0.734 $\pm$ 0.005	0.654 $\pm$ 0.004	0.351 $\pm$ 0.002	0.524 $\pm$ 0.041	N/A	0.774 $\pm$ 0.031
GSC	0.473 $\pm$ 0.000	0.885 $\pm$ 0.000	0.322 $\pm$ 0.000	0.480 $\pm$ 0.000	0.817 $\pm$ 0.000	0.947 $\pm$ 0.000
LMSC	0.737 $\pm$ 0.078	0.765 $\pm$ 0.051	0.357 $\pm$ 0.021	0.573 $\pm$ 0.018	0.938 $\pm$ 0.035	0.917 $\pm$ 0.075
OKMSC	0.789 $\pm$ 0.002	0.843 $\pm$ 0.000	0.416 $\pm$ 0.002	0.546 $\pm$ 0.016	0.948 $\pm$ 0.014	0.959 $\pm$ 0.002

space, probabilistic simplex constraint and graph rank on affinity matrix under one-step framework. By contrast, the comparison methods only consider a part of factors or under two-step framework. Therefore, these methods cannot obtain robust common affinity matrix across multiple views. For example, Fig. 5 shows the common affinity matrix of MLAN, SwMC, GSC and OKMSC on VIS/NIR data set. The clustering results over three evaluation metrics on different multi-view data sets have demonstrated the superior of proposed method.

Next, we follow the work in [66] to statistically verify the advantage of OKMSC over baseline methods. Concretely, t -test [67] with $p < 0.05$ is used for investigating the statistical significance of the difference between our method and the baseline methods. For each data set, we run our method and the baseline methods (include k -means) over 30 times. It is noteworthy that in each

comparison three situations are considered according to t -test: our method is significantly better than the baseline methods/ comparable to the baseline methods/ significantly worse than the baseline methods in terms of three evaluation metrics, i.e., Accuracy (Acc), Normalized Mutual Information (NMI) and F-score, respectively. In particular, the statistical test results are reported in Table 7. From this table, we can see that our method exhibits statistical improvements over the baseline methods on all the six data sets. Especially, for the data sets Caltech101-7, Reuters and VIS/NIR, the statistical test results show that the advantage of our method over the baseline methods is even greater. The above t -test results on six benchmark data sets have further validated the superior of proposed method.

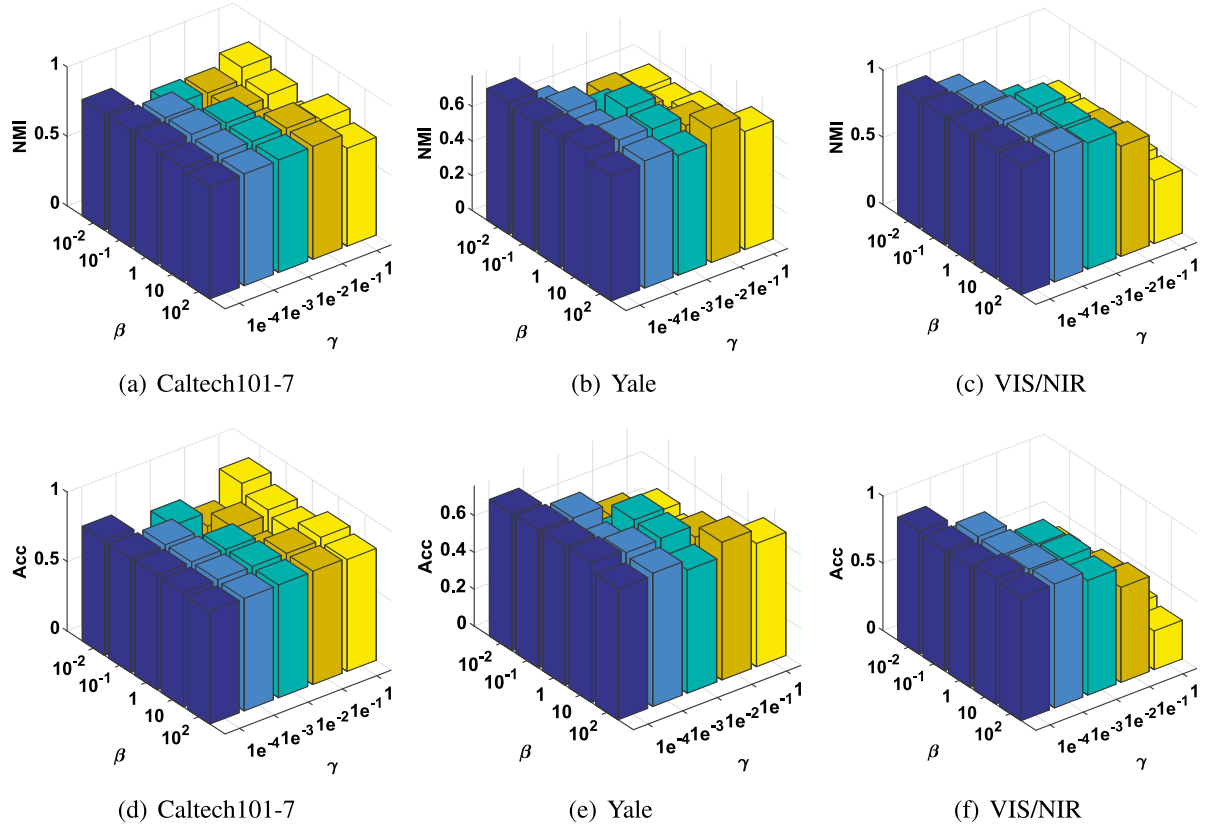


Fig. 6. Analysis on the effect of parameters β and γ by fixing parameter α to 1.

Table 7

The t-test comparison results ($p < 0.05$) of our method over other compared methods in terms of NMI, Acc and F-score on six benchmark data sets. The three integers inside the brackets respectively correspond to the number of times that the performance of our method is (significantly better than, comparable to, and significantly worse than) a baseline method.

Evaluation measures	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
Acc	(15 / 0 / 0)	(14 / 0 / 1)	(12 / 0 / 0)	(11 / 1 / 0)	(8 / 1 / 0)	(10 / 1 / 0)
NMI	(14 / 1 / 0)	(14 / 0 / 1)	(11 / 1 / 0)	(12 / 0 / 0)	(8 / 1 / 0)	(11 / 0 / 0)
F-score	(15 / 0 / 0)	(14 / 0 / 1)	(12 / 0 / 0)	(10 / 1 / 1)	(8 / 1 / 0)	(11 / 0 / 0)
Overall	(44 / 1 / 0)	(42 / 0 / 3)	(35 / 1 / 0)	(33 / 2 / 1)	(24 / 3 / 0)	(32 / 1 / 0)

4.3. Parameter analysis and computational speed analysis

In this subsection, we first investigate the clustering performance of proposed method under different parameter settings. There are three parameters in our method, i.e., α , β and γ . In this experiment, we use the grid strategy to evaluate the effect of these three parameters. Specially, we set the parameters α and β in the same range of $\{0.01, 0.1, 1, 10, 100\}$, while set the parameter γ in the range of $\{1e^{-4}, 1e^{-3}, 1e^{-2}, 1e^{-1}, 1\}$. In this experiment, we analyze the effect of two parameters by fixing one parameter. Three benchmark data sets are used, i.e., Caltech101-7, Yale and VIS/NIR, and the clustering quality is valued in terms of NMI and Acc.

Figs. 6 to 8 plot the parameter effect of α , β and γ on the clustering performance in terms of NMI and Acc, respectively. From these figures, we can observe following two findings. First, for all the data sets, the performance of proposed method is relative insensitive to the parameter β and is more sensitive to the parameter α and γ . Second, when we use the setting of α in the interval of $(10^{-1}, 10)$ as well as the set γ in the interval of $(10^{-3}, 10^{-1})$, the proposed method presents relative promising clustering performance on all the data sets. Therefore, we suggest to set β to a fixed value, e.g., 1, and then use the cross validation

Table 8

Running time (in seconds) of baseline methods on six multi-view data sets.

Method	Caltech101-7	MSRC-v1	Reuters	Yale	Cora	VIS/NIR
S ³ C	72.89	67.25	8406.71	3033.67	N/A	9068.23
TLSC	58.46	4.84	178.18	2.26	2581.12	2253.27
MLAN	0.531	0.521	51.04	0.215	1.54	5.27
SwMC	12.73	0.113	14.11	3.31	20.97	34.55
RMSC	4.42	0.873	5.24	0.378	17.53	23.84
DiMSC	24.98	2.51	37.47	3.54	N/A	80.87
GMC	1.13	0.24	66.87	0.32	1.16	2.52
LMSC	19.63	8.59	75.24	28.72	102.98	128.80
OKMSC	11.56	4.13	35.58	3.27	34.87	45.42

method to find the best combinations of parameters α and γ for specific task.

Next, we investigate the computational speed and the time complexity of proposed method over the compared methods. As shown in Table 8, we can observe that single-view methods do not perform faster than all the multi-view methods. Besides, we can observe that our method is faster than S³C, TLSC, DiMSC and LMSC on most of the multi-view data sets. Furthermore, when compared with the other methods, i.e., MLAN, SwMC and RMSC, our method still achieves comparable performance in terms of

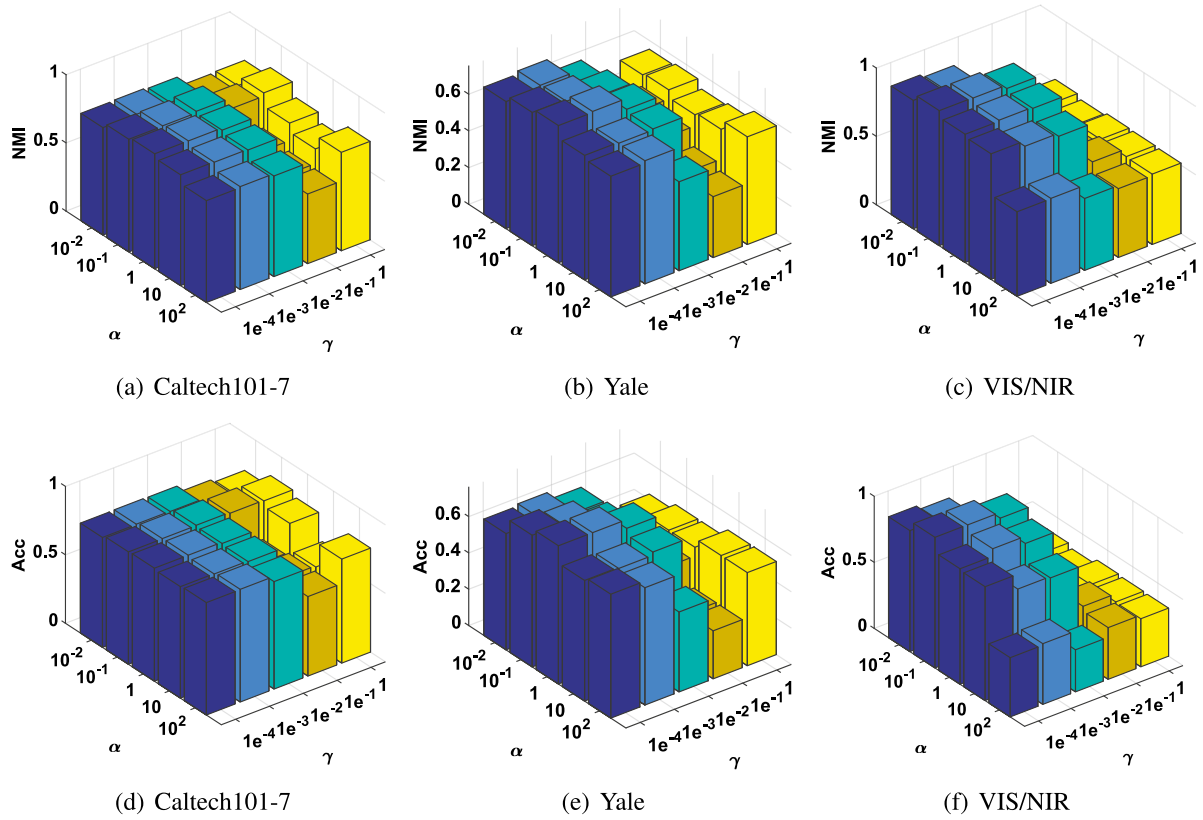


Fig. 7. Analysis on the effect of parameters α and γ by fixing parameter β to 1.

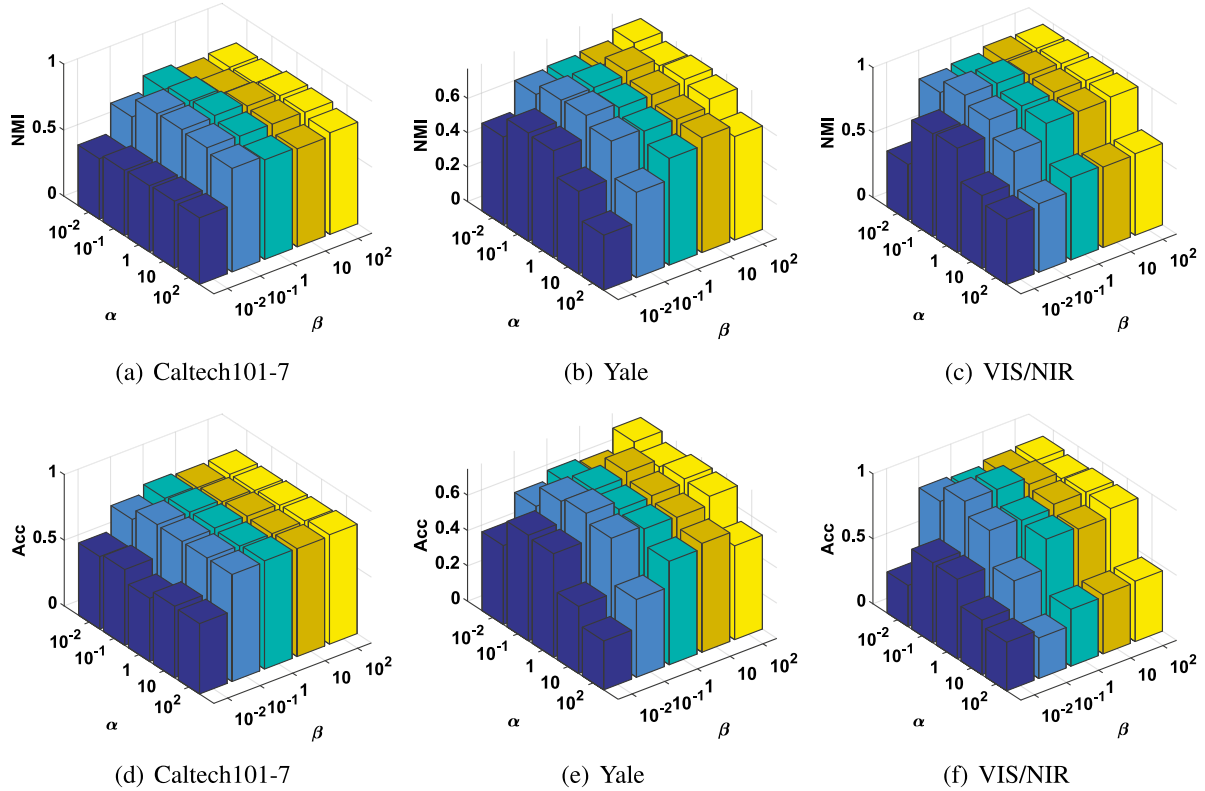


Fig. 8. Analysis on the effect of parameters α and β by fixing parameter γ to 0.01.

computational speed. In particular, Table 9 provides the time complexity of five baseline methods, where N , D , T represent the

number of data samples, the total dimensionality of multi-view data and the iterations, respectively. Especially for S^3C method,

Table 9
Time complexity of some baseline methods.

Method	S ³ C	DiMSC	GMC	LMSC	OKMSC
Time complexity	$O(T_1 T_2 (N^3 + DN^2))$	$O(TN^3 + DN^2)$	$O(TN^2)$	$O(T(D^3 + N^3))$	$O(N^3 + TN^2 \log(N))$

T_1 and T_2 are the number of the inner iterations and the outer iterations, respectively. From this table, we have two aspects of observations. First, it can be observed that the time complexity of S³C and LMSC are relatively expensive over the other compared methods. Second, it can be seen that the computational costs of S³C, DiMSC, and LMSC are not only scaled to the number of sample N , but also related to the dimensionality of data set D . As illustrated in Table 1, note that the condition $N < D$ holds (especially for data sets Reuters and Yale, condition $N \ll D$ holds). Therefore, it is clear to see that the total computational cost of proposed method is less or much less than S³C, DiMSC, and LMSC in our experiments. This is probably the main reason that our method is able to cost less time than the compared methods in most cases (as shown in Table 8).

5. Conclusion

This paper presents a novel one-step framework for multi-view subspace clustering, which aims to address the issues of previous two-step based methods. Unlike the existing multi-view clustering methods, our method learns an optimal common affinity under one-step framework to avoid the suboptimal issue of two-step strategy. Besides, the proposed method extends the linear space to kernel space, so as to capture the nonlinear structure hidden in the multi-view data. Furthermore, to obtain a more robust clustering result, a rank constraint and the simplex probabilistic constraint are introduced to our optimization model, which enables the obtained common affinity matrix with exactly k connected components (where k is the number of clusters). Extensive experimental results on six benchmark data sets have demonstrated the superiority of our method over several popular clustering methods.

Acknowledgments

The authors would like to thank the anonymous reviewers and the Associate Editor for their constructive comments and suggestions, which greatly improve this paper. This work was supported by the NSFC, China (61773410, 61673403, 61976097, 61602189).

References

- [1] M. White, X. Zhang, D. Schuurmans, Y.I. Yu, Convex multi-view subspace learning, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1673–1681.
- [2] Y. Jiang, Z. Deng, F.L. Chung, G. Wang, P. Qian, K.S. Choi, et al., Recognition of epileptic EEG signals using a novel multiview TSK fuzzy system, IEEE Trans. Fuzzy Syst. 25 (1) (2016) 3–20.
- [3] C. Zhang, Y. Liu, Y. Liu, Q. Hu, X. Liu, P. Zhu, FISH-MML: Fisher-HSIC multi-view metric learning, in: Proceedings of the Twenty-Seventh International Joint Conference on Artificial Intelligence, 2018, pp. 3054–3060.
- [4] J. Zhong, N. Wang, Q. Lin, P. Zhong, Weighted feature selection via discriminative sparse multi-view learning, Knowl.-Based Syst. 178 (2019) 132–148.
- [5] X. Zhao, N. Evans, J.L. Dugelay, A subspace co-training framework for multi-view clustering, Pattern Recognit. Lett. 41 (2014) 73–82.
- [6] Y. Liang, D. Huang, C.D. Wang, Consistency meets inconsistency: A unified graph learning framework for multi-view clustering, in: Proceedings of the IEEE International Conference on Data Mining, 2019.
- [7] C. Xu, D. Tao, C. Xu, Multi-view self-paced learning for clustering, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015, pp. 3974–3980.
- [8] F. Nie, G. Cai, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, 2017, pp. 2408–2414.
- [9] K. Zhan, C. Niu, C. Chen, F. Nie, C. Zhang, Y. Yang, Graph structure fusion for multiview clustering, IEEE Trans. Knowl. Data Eng. 31 (2018) 1984–1993.
- [10] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, Knowl.-Based Syst. 158 (2018) 1–8.
- [11] X. Chen, X. Xu, J.Z. Huang, Y. Ye, TW-K-means: automated two-level variable weighting clustering algorithm for multiview data, IEEE Trans. Knowl. Data Eng. 25 (4) (2013) 932–944.
- [12] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in: Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence, 2013, pp. 2598–2604.
- [13] J. Liu, C. Wang, J. Gao, J. Han, Multi-view clustering via joint nonnegative matrix factorization, in: Proceedings of the SIAM International Conference on Data Mining, 2013, pp. 252–260.
- [14] J. Wang, F. Tian, H. Yu, C.H. Liu, K. Zhan, X. Wang, Diverse non-negative matrix factorization for multiview data representation, IEEE Trans. Cybern. 48 (9) (2018) 2620–2632.
- [15] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 393–400.
- [16] R. Xia, Y. Pan, L. Du, J. Yin, Robust multi-view spectral clustering via low-rank and sparse decomposition, in: Proceedings of the Twenty-Eighth AAAI Conference on Artificial Intelligence, 2014, pp. 2149–2155.
- [17] H. Gao, F. Nie, X. Li, H. Huang, Multi-view subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 4238–4246.
- [18] F. Nie, J. Li, X. Li, et al., Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification, in: Proceedings of the Twenty-Five International Joint Conference on Artificial Intelligence, 2016, pp. 1881–1887.
- [19] F. Nie, J. Li, X. Li, Self-weighted multiview clustering with multiple graphs, in: Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, 2017, pp. 2564–2570.
- [20] X. Cao, C. Zhang, H. Fu, S. Liu, H. Zhang, Diversity-induced multi-view subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 586–594.
- [21] C. Zhang, H. Fu, S. Liu, G. Liu, X. Cao, Low-rank tensor constrained multiview subspace clustering, in: Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 1582–1590.
- [22] S. Bickel, T. Scheffer, Multi-view clustering, in: Proceedings of the IEEE International Conference on Data Mining, Vol. 4, 2004, pp. 19–26.
- [23] V.R. De Sa, Spectral clustering with two views, in: Proceedings of the International Conference on Machine Learning Workshop on Learning with Multiple Views, 2005, pp. 20–27.
- [24] G. Tzortzis, A. Likas, Kernel-based weighted multi-view clustering, in: Proceedings of the IEEE International Conference on Data Mining, IEEE, 2012, pp. 675–684.
- [25] Y.M. Xu, C.D. Wang, J.H. Lai, Weighted multi-view clustering with feature selection, Pattern Recognit. 53 (2016) 25–35.
- [26] C.D. Wang, J.H. Lai, S.Y. Philip, Multi-view clustering based on belief propagation, IEEE Trans. Knowl. Data Eng. 28 (4) (2015) 1007–1021.
- [27] G.Y. Zhang, C.D. Wang, D. Huang, W.S. Zheng, Y.R. Zhou, TW-Co-k-means: Two-level weighted collaborative k-means for multi-view clustering, Knowl.-Based Syst. 150 (2018) 127–138.
- [28] S. Huang, Y. Ren, Z. Xu, Robust multi-view data clustering with multi-view capped-norm k-means, Neurocomputing 311 (2018) 197–208.
- [29] Y. Jiang, F.L. Chung, S. Wang, Z. Deng, J. Wang, P. Qian, Collaborative fuzzy clustering from multiple weighted views, IEEE Trans. Cybern. 45 (4) (2014) 688–701.
- [30] S. Huang, Z. Xu, J. Lv, Adaptive local structure learning for document co-clustering, Knowl.-Based Syst. 148 (2018) 74–84.
- [31] P. Qian, J. Zhou, Y. Jiang, F. Liang, K. Zhao, S. Wang, et al., Multi-view maximum entropy clustering by jointly leveraging inter-view collaborations and intra-view-weighted attributes, IEEE Access 6 (2018) 28594–28610.
- [32] L. Huang, H.Y. Chao, C.D. Wang, Multi-view intact space clustering, Pattern Recognit. 86 (2019) 344–353.
- [33] Z. Zhang, L. Liu, F. Shen, H.T. Shen, L. Shao, Binary multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. 41 (7) (2018) 1774–1782.
- [34] J. He, R. Lawrence, A graphbased framework for multi-task multi-view learning, in: Proceedings of the 28th International Conference on Machine Learning, 2011, pp. 25–32.

- [35] Y. Zhang, Y. Yang, T. Li, H. Fujita, A multitask multiview clustering algorithm in heterogeneous situations based on LLE and LE, *Knowl.-Based Syst.* 163 (2019) 776–786.
- [36] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *science* 290 (5500) (2000) 2323–2326.
- [37] M. Belkin, P. Niyogi, Laplacian Eigenmaps and spectral techniques for embedding and clustering, in: *Proceedings of the Advances in Neural Information Processing Systems*, 2002, pp. 585–591.
- [38] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: *Proceedings of the Advances in Neural Information Processing Systems*, 2011, pp. 1413–1421.
- [39] K. Zhan, C. Zhang, J. Guan, J. Wang, Graph learning for multiview clustering, *IEEE Trans. Cybern.* (99) (2017) 1–9.
- [40] S. Huang, Z. Kang, I.W. Tsang, Z. Xu, Auto-weighted multi-view clustering via kernelized graph learning, *Pattern Recognit.* 88 (2019) 174–184.
- [41] C. Zhang, H. Fu, Q. Hu, X. Cao, Y. Xie, D. Tao, D. Xu, Generalized latent multi-view subspace clustering, *IEEE Trans. Pattern Anal. Mach. Intell.* (2018).
- [42] E. Elhamifar, R. Vidal, Sparse subspace clustering: Algorithm, theory, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11) (2013) 2765–2781.
- [43] C.-Y. Lu, H. Min, Z.-Q. Zhao, L. Zhu, D.-S. Huang, S. Yan, Robust and efficient subspace segmentation via least squares regression, in: *Proceedings of the European Conference on Computer Vision*, 2012, pp. 347–360.
- [44] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2013) 171–184.
- [45] X. Zhu, S. Zhang, R. Hu, W. He, C. Lei, P. Zhu, One-step multi-view spectral clustering, *IEEE Trans. Knowl. Data Eng.* 31 (2018) 2022–2034.
- [46] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 2014, pp. 977–986.
- [47] K. Fan, On a theorem of weyl concerning eigenvalues of linear transformations i, *Proc. Natl. Acad. Sci.* 35 (11) (1949) 652–655.
- [48] S. Boyd, N. Parikh, E. Chu, B. Peleato, J. Eckstein, et al., Distributed optimization and statistical learning via the alternating direction method of multipliers, *Found. Trends[®] Mach. Learn.* 3 (1) (2011) 1–122.
- [49] B. Liu, Y. Li, Z. Xu, Manifold regularized matrix completion for multi-label learning with ADMM, *Neural Netw.* 101 (2018) 57–67.
- [50] J. Duchi, S. Shalev-Shwartz, Y. Singer, T. Chandra, Efficient projections onto the l_1 -ball for learning in high dimensions, in: *Proceedings of the 25th International Conference on Machine Learning*, ACM, 2008, pp. 272–279.
- [51] D. Huang, C.-D. Wang, J. Wu, J.-H. Lai, C.K. Kwok, Ultra-scalable spectral clustering and ensemble clustering, *IEEE Trans. Knowl. Data Eng.* (2019).
- [52] Y. Li, F. Nie, H. Huang, J. Huang, Large-scale multi-view spectral clustering via bipartite graph, in: *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, 2015, pp. 2750–2756.
- [53] M. Wang, W. Fu, S. Hao, D. Tao, X. Wu, Scalable semi-supervised learning by efficient anchor graph regularization, *IEEE Trans. Knowl. Data Eng.* 28 (7) (2016) 1864–1877.
- [54] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, Y. Ma, Robust recovery of subspace structures by low-rank representation, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (1) (2012) 171–184.
- [55] S. Zhan, J. Wu, N. Han, J. Wen, X. Fang, Unsupervised feature extraction by low-rank and sparsity preserving embedding, *Neural Netw.* 109 (2019) 56–66.
- [56] J. Wen, N. Han, X. Fang, L. Fei, K. Yan, S. Zhan, Low-rank preserving projection via graph regularized reconstruction, *IEEE Trans. Cybern.* 49 (4) (2018) 1279–1291.
- [57] C. Hou, F. Nie, H. Tao, D. Yi, Multi-view unsupervised feature selection with adaptive similarity and view weight, *IEEE Trans. Knowl. Data Eng.* 29 (9) (2017) 1998–2011.
- [58] H. Zhao, Z. Ding, Y. Fu, Multi-view clustering via deep matrix factorization, in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 2921–2927.
- [59] S. Zhong, J. Ghosh, A unified framework for model-based clustering, *J. Mach. Learn. Res.* 4 (Nov) (2003) 1001–1037.
- [60] K.-Y. Lin, L. Huang, C.-D. Wang, H.-Y. Chao, Multi-view proximity learning for clustering, in: *Proceedings of the International Conference on Database Systems for Advanced Applications*, Springer, 2018, pp. 407–423.
- [61] C.-G. Li, R. Vidal, Structured sparse subspace clustering: A unified optimization framework, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 277–286.
- [62] Z. Kang, C. Peng, Q. Cheng, Twin learning for similarity and clustering: A unified kernel approach, in: *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, 2017, pp. 2080–2086.
- [63] H. Wang, Y. Yang, B. Liu, H. Fujita, A study of graph-based system for multi-view clustering, *Knowl.-Based Syst.* 163 (2019) 1009–1019.
- [64] C. Zhang, Q. Hu, H. Fu, P. Zhu, X. Cao, Latent multi-view subspace clustering, in: *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017, pp. 4279–4287.
- [65] F. Nie, X. Wang, M.I. Jordan, H. Huang, The constrained Laplacian rank algorithm for graph-based clustering, in: *Proceedings of the Thirtieth AAAI Conference on Artificial Intelligence*, 2016, pp. 1969–1976.
- [66] D. Huang, C.-D. Wang, J.-H. Lai, Locally weighted ensemble clustering, *IEEE Trans. Cybern.* 48 (5) (2017) 1460–1473.
- [67] J. Demšar, Statistical comparisons of classifiers over multiple data sets, *J. Mach. Learn. Res.* 7 (Jan) (2006) 1–30.