Pattern Recognition 88 (2019) 174–184

Contents lists available at ScienceDirect

Pattern Recognition

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

Auto-weighted multi-view clustering via kernelized graph learning

Shudong Huang[a], Zhao Kang[a], Ivor W. Tsang[b], Zenglin Xu[a,*]

[a] SMILE Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China
[b] Centre for Artificial Intelligence, University of Technology Sydney, NSW 2007, Australia

ARTICLE INFO

Article history:
Received 1 March 2018
Revised 24 September 2018
Accepted 10 November 2018
Available online 22 November 2018

Keywords:
Graph learning
Multi-view clustering
Multiple kernel learning
Auto-weighted strategy

ABSTRACT

Datasets are often collected from different resources or comprised of multiple representations (i.e., views). Multi-view clustering aims to analyze the multi-view data in an unsupervised way. Owing to the efficiency of uncovering the hidden structures of data, graph-based approaches have been investigated widely for various multi-view learning tasks. However, similarity measurement in these methods is challenging since the construction of similarity graph is impacted by several factors such as the scale of data, neighborhood size, choice of similarity metric, noise and outliers. Moreover, nonlinear relationships usually exist in real-world datasets, which have not been considered by most existing methods. In order to address these challenges, a novel model which simultaneously performs multi-view clustering task and learns similarity relationships in kernel spaces is proposed in this paper. The target optimal graph can be directly partitioned into exact c connected components if there are c clusters. Furthermore, our model can assign ideal weight for each view automatically without additional parameters as previous methods do. Since the performance is often sensitive to the input kernel matrix, the proposed model is further extended with multiple kernel learning ability. With the proposed joint model, three subtasks including construct the most accurate similarity graph, automatically allocate optimal weight for each view and find the cluster indicator matrix can be simultaneously accomplished. By this joint learning, each subtask can be mutually enhanced. Experimental results on benchmark datasets demonstrate that our model outperforms other state-of-the-art multi-view clustering algorithms.

1. Introduction

Clustering is one of the most important topic in machine learning [1–3]. Nowadays, more and more datasets are represented by different views where the encoded information is complementary to each other. Thus it is critical to designing algorithms to fuse these heterogeneous features such that the accuracy and robustness can be improved. For graph-based algorithms, if we use a graph to denote the relation of entities in each view, then multiple graphs can be naturally obtained. Therefore, it is clear that we can transform the graph-based multi-view clustering into a multiple graph learning problem.

Multi-view clustering recently provides an important solution for analyzing the multi-view data. Unlike the traditional methods which rely on single view [4–6], multi-view clustering tends to fuse the compatible and complementary information hidden in the views [7–13]. In the past decade, several methods have been proposed to solve the multi-view clustering problem. In general, existing multi-view clustering methods can be roughly classified into two categories including subspace approaches which aim to explore the hidden common subspace shared by all views [14–19], and graph-based methods which were designed based on the traditional spectral learning models [20–24]. However, most existing multi-view clustering methods still suffer from the following drawbacks: (1) These methods are sensitive to outliers and noisy data, thus impair the final clustering performance greatly; (2) Nonlinear relationships exist in real-world dataset have not been effectively searched by most existing methods. (3) Different kernels applied in graph-based methods usually lead to distinct results. It is a great challenge to obtain robust performance with different kernels [25].

In this paper, a novel model which simultaneously performs multi-view clustering task and learns similarity information in kernel spaces is proposed. The target optimal graph can be directly partitioned into exact c connected components which is exactly equal to the cluster number. Meanwhile, the proposed model can assign ideal weight for each view automatically without additional parameters as previous methods do. A multiple kernel learning approach is further developed to tackle the practical problem of how to choose the most suitable kernel. With the proposed joint model, we can simultaneously accomplish three subtasks of constructing the most accurate similarity graph, allocating ideal weight for each

* Corresponding author.
E-mail addresses: zlxu@uestc.edu.cn, zenglin@gmail.com (Z. Xu).

https://doi.org/10.1016/j.patcog.2018.11.007
0031-3203/

view automatically and finding the cluster indicator matrix. By this joint learning, each subtask can be mutually enhanced.

2. Related work

As one of the most important part of multi-view clustering methods, namely, multi-view spectral clustering, has been widely investigated. Kumar and Daumé presented a novel multi-view spectral clustering method by incorporating the strategy of co-training [21]. It is based on the assumption that an instance should belong to the same cluster with respect to different views according to the true underlying clustering. In [22], Kumar et al. further proposed another novel multi-view spectral clustering method named co-regularized multi-view spectral clustering. This method allows eigenvectors of each views to look similar by enforcing these eigenvectors towards a specific common consensus.

There are also several subspace-based multi-view clustering methods proposed by now. Gao et al. presented a novel method named t-SVD multi-view clustering (MVSC) [26]. MVSC aims to simultaneously perform clustering on subspace representations of each view. Based on the assumption that feature selection can improving the performance of multi-view clustering methods, Xu et al. [27] proposed the weighted multi-view clustering which simultaneously performs the feature selection and multi-view data clustering. Tzortzis and Likas [28] presented a kernel-based multi-view clustering, namely, multi-view kernel clustering (MVKKM). Each view in MVKKM is represented as a given kernel, and all kernels are processed in parallel with the weighted combination strategy. Cai et al. [29] presented a robust multi-view clustering based on the traditional K-means clustering algorithm (RMKMC). With the help of $l_{2,1}$-norm, RMKMC shows a strong ability of robustness when dealing with outliers. Motivated by the max-product belief propagation, Wang et al. [30] proposed a belief propagation multi-view clustering algorithm (BPMVC). The goal of BPMVC is to build multi-view clustering model which consists of two components. Zong et al. [31] presented a multi-manifold regularized multi-view clustering which incorporates consensus manifold regularization such that the local geometrical structure of all views can be preserved.

3. Problem formulation

We propose a novel model which performs multi-view clustering task and learns similarity relationships in kernel spaces simultaneously. To enhance the robustness of our model w.r.t different kernels, we extend our model such that it has a multiple kernel learning ability. And finally, we propose another model named auto-weighted multi-view clustering with multiple kernels.

Notations. In this paper, we use boldface uppercase letters to denote the matrices. For a matrix $\mathbf{A}$, boldface lowercase letter $\mathbf{a}_i$ and lowercase letter a_{ij} denote the ith column and ijth element of $\mathbf{A}$. Tr($\mathbf{A}$) represents the trace of $\mathbf{A}$, while $\mathbf{A}$ denotes the Frobenius norm of $\|\mathbf{A}\|_F$. $\mathbf{A}\geq 0$ means that all elements of $\mathbf{A}$ are equal to or larger than zero.

3.1. Background and motivation

Given a data matrix $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_n]\in\mathbb{R}^{k\times n}$ with n data points and k features. According to the self-expressive property [32,33], each data point can be represented as linear combination of other points. This can be formulated as

$$\begin{aligned}&\mathbf{x}_i=\sum_j \mathbf{x}_j s_{ij}\\ &s.t.\quad \mathbf{S}\geq 0,\end{aligned}\tag{1}$$

where $\mathbf{S}=[s_{ij}]\in\mathbb{R}^{n\times n}$ and s_{ij} denotes the weight between the jth and ith data point. $\mathbf{S}$ can be treated as the similarity matrix. It represents the global structure of data and can be obtained by solving

$$\begin{aligned}&\min_{\mathbf{S}}\|\mathbf{X}-\mathbf{X}\mathbf{S}\|_F^2+\mu\|\mathbf{S}\|_F^2\\ &s.t.\quad \mathbf{S}\geq 0,\end{aligned}\tag{2}$$

where μ is a balancing parameter. It can be seen that Eq. (2) assumes linear relations between data points. Since the nonlinear relations between the data points exist widely in real-world datasets, thus it is critical to recover the nonlinear relations. Inspired by the general kernelization framework [33,34], Eq. (2) can be extended to kernel spaces such that the nonlinear relations between instances can be explored. In detail, suppose $\Phi:\mathcal{R}^D\rightarrow\mathcal{H}$ denotes a kernel which is obtained by mapping the data points from the original data space to the reproducing Hilbert space $\mathcal{H}$. For a input data matrix $\mathbf{X}=[\mathbf{x}_1,\ldots,\mathbf{x}_n]$, the transformation is $\Phi(\mathbf{X})=[\Phi(\mathbf{x}_1),\ldots,\Phi(\mathbf{x}_n)]$. A prescribed kernel $\mathbf{K}_{\mathbf{x}_i,\mathbf{x}_j}=<\Phi(\mathbf{x}_i),\Phi(\mathbf{x}_j)>$ is used to capture the similarity between $\mathbf{x}_i$ and $\mathbf{x}_j$. By virtue of the kernel trick, there is no need to verify the specific transformation Φ. Thus it greatly simplifies the computations. We rewrite Eq. (2) as

$$\begin{aligned}&\min_{\mathbf{S}}\mathrm{Tr}\left(\mathbf{K}-2\mathbf{K}\mathbf{S}+\mathbf{S}^T\mathbf{K}\mathbf{S}\right)+\mu\|\mathbf{S}\|_F^2\\ &s.t.\quad \mathbf{S}\geq 0.\end{aligned}\tag{3}$$

By solving Eq. (3), we capture the linear relations brought by $\Phi(X)$, as well as the nonlinear relations in the original dataset. Note that Eq. (3) becomes Eq. (2) if a linear kernel is applied.

3.2. Multi-view clustering with single kernel

The formulations mentioned above are presented for solving the single view clustering problems. Since many real world datasets are collected from different views, it is essential to extend the single models to solve the multi-view application problems. For a multi-view data, denote $\mathbf{X}^{(1)},\ldots,\mathbf{X}^{(m)}$ as the data matrix with m views. $\mathbf{X}^{(v)}=[\mathbf{x}_1^{(v)},\ldots,\mathbf{x}_n^{(v)}]\in\mathbb{R}^{n\times k^{(v)}}$ is the vth view with $k^{(v)}$ features. For each view, an individual kernel can be constructed, thus we have $\mathbf{K}^{(1)},\ldots,\mathbf{K}^{(m)}\in\mathbb{R}^{n\times n}$. For the vth kernel, we have $\mathbf{K}^{(v)}_{\mathbf{x}_i^{(v)},\mathbf{x}_j^{(v)}}=<\Phi(\mathbf{x}_i^{(v)}),\Phi(\mathbf{x}_j^{(v)})>$. Obviously, a straightforward way is to stack up these kernels to a new one. And then put it into the classical spectral model. However, the importance of different kernels is ignored in this way and may suffer if unreliable kernel is considered. Instead, we can combine these different kernels linearly with suitable weights $w_v(v=1,\ldots,m)$, and introduce a parameter γ to keep the smooth of the weights distribution. We force the similarity matrix $\mathbf{S}$ to be unified for all views. Thus, this thought can be formulated as

$$\begin{aligned}&\min_{\mathbf{S},w_v}\sum_v (w_v)^\gamma\mathrm{Tr}\left(\mathbf{K}^{(v)}-2\mathbf{K}^{(v)}\mathbf{S}+\mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)+\mu\|\mathbf{S}\|_F^2\\ &s.t.\quad \mathbf{S}\geq 0,0\leq w_v\leq 1,w_v^T\mathbf{1}=1,\end{aligned}\tag{4}$$

where γ is the non-negative scalar to control the weights distribution, and it could be regularization parameter in another model:

$$\begin{aligned}&\min_{\mathbf{S},w_v}\sum_v\left(w_v\mathrm{Tr}\left(\mathbf{K}^{(v)}-2\mathbf{K}^{(v)}\mathbf{S}+\mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)+\gamma\|w_v\|_2^2\right)\\ &\quad+\mu\|\mathbf{S}\|_F^2\\ &s.t.\quad \mathbf{S}\geq 0,0\leq w_v\leq 1,w_v^T\mathbf{1}=1.\end{aligned}\tag{5}$$

It is well known that the fewer parameters to be tuned, the more robustness learning algorithms possess, especially for unsupervised learning models. We can see γ can be searched in a large range.

Since the choice of γ is crucial to the multi-view clustering performance and its ideal value varies for different datasets, it is really elusive to preserve good performance and at the same time rely less on parameter searching. Thus, we expect to remove such a parameter while pursuing good performance. In the next section, we will present an auto-weighted strategy to alleviate such challenging problem.

3.3. Auto-weighted multiple graph learning in kernel space

Motivated by recently proposed *Iteratively Re-weighted* technique [35,36], we propose a novel model for multiple graph learning in kernel space:

$$\min_{\mathbf{S}} \sum_{v} \sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2 \quad s.t. \quad \mathbf{S} \geq 0. \tag{6}$$

We can see there is no weight factors explicitly defined in Eq. (6). Let Λ be the Lagrange multiplier, the Lagrange function of Eq. (6) can be written as

$$\sum_{v} \sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2 + \Gamma(\Lambda, \mathbf{S}), \tag{7}$$

where $\Gamma(\Lambda, \mathbf{S})$ denotes the formalized term derived from constraints. If we take the derivative of Eq. (7) w.r.t $\mathbf{S}$ and set the derivative to zero, then we obtain

$$\sum_{v} w_v \frac{\partial \mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)}{\partial \mathbf{S}} + \frac{\mu\partial\|\mathbf{S}\|_F^2}{\partial \mathbf{S}} + \frac{\partial\Gamma(\Lambda, \mathbf{S})}{\partial \mathbf{S}} = 0, \tag{8}$$

where

$$w_v = 1 \Big/ \left(2\sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)}\right). \tag{9}$$

As shown in Eq. (10), w_v is dependent on $\mathbf{S}$, thus Eq. (9) cannot be solved directly. But when w_v is considered to be stationary, Eq. (9) can be considered as a solution for problem

$$\min_{\mathbf{S}} \sum_{v} w_v \mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right) + \mu\|\mathbf{S}\|_F^2 \quad s.t. \quad \mathbf{S} \geq 0. \tag{10}$$

Based on the assumption that w_v is stationary, the Lagrange function of Eq. (6) also apply to Eq. (11). After we calculate $\mathbf{S}$ from Eq. (11), w_v can also be updated correspondingly, which suggests optimizing Eq. (6) in an alternative way.

Furthermore, we expect that the target graph can be directly partitioned into exact c connected components if there are c clusters in the given dataset. We can use another important property of Laplacian matrix, which is stated as [37].

Theorem 1. *The multiplicity c of the eigenvalue 0 of the Laplacian matrix $\mathbf{L}$ is equal to the number of connected components in the graph with the similarity matrix $\mathbf{S}$.*

Theorem 1 denotes that if the constraint $rank(\mathbf{L}) = n - c$ satisfies, then $\mathbf{S}$ contains exactly c connected components. With this constraint, our model in Eq. (11) can be reformulated as

$$\min_{\mathbf{S}} \sum_{v} w_v \mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right) + \mu\|\mathbf{S}\|_F^2 \quad s.t. \quad \mathbf{S} \geq 0, rank(\mathbf{L}) = n - c, \tag{11}$$

where $\mathbf{L} = \mathbf{D} - \frac{\mathbf{S}^T + \mathbf{S}}{2}$ and $\mathbf{D}$ is a diagonal matrix and its ith diagonal element is denoted as $\sum_j \frac{s_{ij} + s_{ji}}{2}$. Let $\sigma_i(L)$ be the ith smallest eigenvalue of $\mathbf{L}$. $\sigma_i(L) \geq 0$ since $\mathbf{L}$ is positive semi-definite. Then $rank(\mathbf{L}) = n - c$ will be satisfied if $\sum_{i=1}^{c} \sigma_i(L) = 0$. Thus we can reformulate the model as

$$\min_{\mathbf{S}} \sum_{v} w_v \mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right) + \mu\|\mathbf{S}\|_F^2 + \lambda\sum_{i=1}^{c}\sigma_i(L) \quad s.t. \quad \mathbf{S} \geq 0. \tag{12}$$

When λ is large enough, the minimization will force regularizer $\sum_{i=1}^{c} \sigma_i(L)$ approaches to 0. Thus $rank(\mathbf{L}) = n - c$ could be satisfied. Eq. (13)is still not easy to solve due to the last term. Fortunately, according to the Ky Fan's Theorem [38], we obtain

$$\sum_{i=1}^{c} \sigma_i(L) = \min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \mathrm{Tr}\left(\mathbf{P}^T\mathbf{L}\mathbf{P}\right), \tag{13}$$

where $\mathbf{P} \in \mathbb{R}^{n \times c}$ is the cluster indicator matrix. Finally, our auto-weighted multi-view clustering with single kernel (MVCSK) can be formulated as

$$\min_{\mathbf{S},\mathbf{P}} \sum_{v} w_v \mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right) + \mu\|\mathbf{S}\|_F^2 + \lambda\mathrm{Tr}\left(\mathbf{P}^T\mathbf{L}\mathbf{P}\right) \quad s.t. \quad \mathbf{S} \geq 0, \mathbf{P}^T\mathbf{P} = \mathbf{I}. \tag{14}$$

3.4. Optimization algorithm of MVCSK

We present an efficient updating algorithm to solve the optimization problem in Eq. (15). Specifically, the objective will be optimized w.r.t one variable while fix other variables. And the procedure repeats until convergence.

(1) Updating $\mathbf{P}$ with fixed w_v and $\mathbf{S}$: The first and second item of Eq. (15) can be considered as constant when w_v and $\mathbf{S}$ are fixed, then optimizing Eq. (15) w.r.t. $\mathbf{P}$ is equivalent to optimizing

$$\min_{\mathbf{P}^T\mathbf{P}=\mathbf{I}} \mathrm{Tr}\left(\mathbf{P}^T\mathbf{L}\mathbf{P}\right). \tag{15}$$

We can obtain the optimal solution $\mathbf{P}$ by calculating the c eigenvectors of $\mathbf{L}$ corresponding to the c smallest eigenvalues.

(2) Updating $\mathbf{S}$ with fixed w_v and $\mathbf{P}$: Eq. (15) can be written column-wisely as

$$\min_{\mathbf{S}} \sum_{v} w_v\left(\mathbf{k}_{ii}^{(v)} - 2\mathbf{k}_{i,:}^{(v)}\mathbf{s}_i + \mathbf{s}_i^T\mathbf{K}^{(v)}\mathbf{s}_i\right) + \mu\mathbf{s}_i^T\mathbf{s}_i + \frac{\lambda}{2}d_i^T\mathbf{s}_i \quad s.t. \quad s_{ij} \geq 0, \tag{16}$$

where $d_i \in \mathbb{R}^{n \times 1}$ denotes vector with the jth element $d_{ij} = \|\mathbf{p}_{i,:} - \mathbf{p}_{j,:}\|^2$. To obtain Eq. (17), the following important equation in spectral analysis is used

$$\frac{1}{2}\sum_{ij}\|\mathbf{p}_{i,:} - \mathbf{p}_{j,:}\|^2 s_{ij} = \mathrm{Tr}(\mathbf{P}^T\mathbf{L}\mathbf{P}). \tag{17}$$

Eq. (17) can be further simplified as

$$\min_{\mathbf{s}_i} \mathbf{s}_i^T\left(\sum_{v} w_v\mathbf{K}^{(v)} + \mu\mathbf{I}\right)\mathbf{s}_i + \left(\frac{\lambda d_i^T}{2} - 2\sum_{v} w_v\mathbf{k}_{i,:}^{(v)}\right)\mathbf{s}_i \quad s.t. \quad \mathbf{s}_i \geq 0. \tag{18}$$

We can obtain $\mathbf{s}_i$ by setting the derivative of Eq. (19) w.r.t. $\mathbf{s}_i$ to be zero.

(3) Updating w_v with fixed $\mathbf{S}$ and $\mathbf{P}$: w_v can be updated by using Eq. (10).

The details of the proposed MVCSK optimization are given in Algorithm 2.

Algorithm 1 Algorithm of MVCSK.

Input: Predefined kernel matrices $\mathbf{K}^{(v)}$ for each view;
Parameters μ and λ;
The number of clusters c.

Output: Similarity matrix $\mathbf{S}$ with exact c connected components.

1: Initialize $\mathbf{S}$ to identity matrix.
2: **repeat**
3: Update $\mathbf{P}$, which is formed by the c eigenvectors of $\mathbf{L}$ corresponding to the c smallest eigenvalues.
4: For each i, update the i-th column of $\mathbf{S}$ by solving Eq. (18) and $\mathbf{S} = max(\mathbf{S}, 0)$.
5: Update w_v by using Eq. (9).
6: **until** Meet the terminating condition

Algorithm 2 Algorithm of MVCMK.

Input: Predefined kernel matrices $\{\mathbf{K}_i^{(v)}\}_{i=1}^r$;
Parameters α, μ and λ;
The number of clusters c.

Output: Similarity matrix $\mathbf{S}$ with exact c connected components.

1: Initialize $\mathbf{S}$ to identity matrix and parameter $\alpha = \frac{1}{r}$.
2: **repeat**
3: Calculate $\widehat{\mathbf{K}}^{(v)}$ as defined in Eq. (24).
4: Do steps 3-5 in Algorithm 1.
5: Calculate $\mathbf{q}$ by using Eq. (28).
6: Update α by solving Eq. (30).
7: **until** Meet the terminating condition

3.5. Convergence analysis

In this section, we prove the convergence of the Algorithm 2. Before we prove its convergence, first we introduce an important lemma as follows [39]:

Lemma 1. *For any positive real number q and t, the following inequality holds:*

$$\sqrt{q} - \frac{q}{2\sqrt{t}} \leq \sqrt{t} - \frac{t}{2\sqrt{t}}. \tag{19}$$

Theorem 1. *Updated* $\mathbf{S}$ *will monotonically decrease the objective in Eq. (12), which makes a solution converge to the local optimum of Eq. (12).*

Proof. Suppose the alternatively updated $\mathbf{S}$ is $\widetilde{\mathbf{S}}$ in each iteration. According to the first step of loop in Algorithm 2, we have

$$\widetilde{\mathbf{S}} = \underset{\mathbf{S}\geq 0, rank(\mathbf{L})=n-c}{\arg\min} \sum_v w_v \mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right) + \mu\|\mathbf{S}\|_F^2 \tag{20}$$

Combining with Eq. (10), i.e., $w_v = \frac{1}{2\sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)}}$, we have

$$\sum_v \frac{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\widetilde{\mathbf{S}} + \widetilde{\mathbf{S}}^T\mathbf{K}^{(v)}\widetilde{\mathbf{S}}\right) + \mu\|\widetilde{\mathbf{S}}\|_F^2}{2\sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2} \leq \sum_v \frac{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right) + \mu\|\mathbf{S}\|_F^2}{2\sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2}. \tag{21}$$

According to Lemma 1, we have

$$\begin{aligned}
&\sum_v \sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\widetilde{\mathbf{S}} + \widetilde{\mathbf{S}}^T\mathbf{K}^{(v)}\widetilde{\mathbf{S}}\right)} + \mu\|\widetilde{\mathbf{S}}\|_F^2 \\
&- \sum_v \frac{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\widetilde{\mathbf{S}} + \widetilde{\mathbf{S}}^T\mathbf{K}^{(v)}\widetilde{\mathbf{S}}\right) + \mu\|\widetilde{\mathbf{S}}\|_F^2}{2\sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2} \\
&\leq \sum_v \sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2 \\
&- \sum_v \frac{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right) + \mu\|\mathbf{S}\|_F^2}{2\sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2}.
\end{aligned} \tag{22}$$

By summing over Eq. (22) and Eq. (23) in the two sides, we obtain

$$\begin{aligned}
&\sum_v \sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\widetilde{\mathbf{S}} + \widetilde{\mathbf{S}}^T\mathbf{K}^{(v)}\widetilde{\mathbf{S}}\right)} + \mu\|\widetilde{\mathbf{S}}\|_F^2 \\
&\leq \sum_v \sqrt{\mathrm{Tr}\left(\mathbf{K}^{(v)} - 2\mathbf{K}^{(v)}\mathbf{S} + \mathbf{S}^T\mathbf{K}^{(v)}\mathbf{S}\right)} + \mu\|\mathbf{S}\|_F^2,
\end{aligned} \tag{23}$$

which completes the prove. □

4. Auto-weighted multi-view clustering with multiple kernels

Although our model described in Eq. (15) can automatically learn the cluster indicator matrix, similarity relations and the optimal weight for each view, the performance of MVCSK is still extremely influenced by the choice of kernel. Furthermore, real-world datasets, especially multi-view datasets, are usually collected from different sources. It is obvious that the different and complementary information of these datasets cannot be fully utilized by single kernel method. Multiple kernel learning provides a natural formulation to integrate complementary information and identify suitable kernel for a specific task. In this section, a novel method is proposed to learn an appropriate consensus kernel from a convex combination of several predefined kernel matrices.

Given r different kernel functions. For the vth view of a multi-view dataset, let $\{\mathbf{K}_i^{(v)}\}_{i=1}^r$ be the r different kernel matrices. Correspondingly, for the vth view, there would be r different kernel spaces $\{\mathcal{H}_i^{(v)}\}_{i=1}^r$. Then we can construct an augmented Hilbert space, $\widehat{\mathcal{H}}^{(v)} = \bigoplus_{i=1}^r \mathcal{H}_i^{(v)}$, by utilizing the mapping of $\widehat{\Phi}(\mathbf{x}) = [\sqrt{\alpha_1}\Phi_1(\mathbf{x}), \ldots, \sqrt{\alpha_r}\Phi_r(\mathbf{x})]^T$ with different weights

Table 1
Description of the datasets (dimensionality).

View	Texas	Cornell	Washington	Winconsin	BBC	BBCSport	NUS
1	Content (1703)	Content (1703)	Content (1703)	Content (1703)	Segment1 (5470)	Segment1 (1991)	Color Histogram (65)
2	Citation (187)	Citation (195)	Citation (230)	Citation (265)	Segment2 (5549)	Segment2 (2063)	Color Moments (226)
3	–	–	–	–	Segment3 (5483)	Segment3 (2113)	Color Correlation (145)
4	–	–	–	–	–	Segment4 (2158)	Edge Distribution (74)
5	–	–	–	–	–	–	Wavelet Texture (129)
Data points	187	195	230	265	1268	116	11576
Classes	5	5	5	5	5	2	15

Table 2
Clustering performance on Texas (%).

Method	ACC	Purity	NMI
SC(1)	48.56(3.62)	56.68(1.65)	14.41(2.88)
SC(2)	38.40(2.05)	59.14(0.29)	11.17(1.95)
Co-train	49.63(1.53)	58.18(0.24)	14.75(1.55)
Co-reg	54.55(0.00)	56.15(0.00)	13.75(0.00)
MVKKM	48.34(6.93)	56.68(0.85)	9.67(3.75)
RMKMC	58.18(2.71)	60.32(2.37)	20.11(3.17)
AMGL	55.94(0.48)	56.79(0.24)	15.11(0.62)
MLAN	55.61(0.00)	63.98(0.00)	18.12(0.00)
MVCSK	**66.84(0.00)**	**71.98(0.00)**	**21.67(0.00)**
MVCMK	**61.50(0.00)**	**65.78(0.00)**	**29.92(0.00)**

$\sqrt{\alpha_i}(\alpha_i \geq 0)$. According to [40], we can denote the consensus kernel of the vth view as

$$\widehat{\mathbf{K}}^{(v)}(\mathbf{x},\mathbf{y}) = < \widehat{\Phi}(\mathbf{x}), \widehat{\Phi}(\mathbf{y}) > = \sum_{i=1}^{r} \alpha_i \mathbf{K}_i^{(v)}. \tag{24}$$

With the convex combination of the positive semidefinite kernel matrices $\{\mathbf{K}_i^{(v)}\}_{i=1}^{r}$, the consensus kernel $\widehat{\mathbf{K}}^{(v)}$ obtained in Eq. (25) is still a positive semidefinite kernel matrix. Thus $\widehat{\mathbf{K}}^{(v)}$ satisfies Mercer's condition [33]. Finally, our auto-weighted multi-view clustering with multiple kernels (MVCMK) can be formulated as

$$\begin{aligned} &\min_{\mathbf{S},\mathbf{P},\alpha} \sum_{v} w_v \mathrm{Tr}\left(\widehat{\mathbf{K}}^{(v)} - 2\widehat{\mathbf{K}}^{(v)}\mathbf{S} + \mathbf{S}^T\widehat{\mathbf{K}}^{(v)}\mathbf{S}\right) \\ &\quad + \mu\|\mathbf{S}\|_F^2 + \lambda \mathrm{Tr}\left(\mathbf{P}^T\mathbf{L}\mathbf{P}\right) \\ &\quad s.t. \quad \mathbf{S} \geq 0, \mathbf{P}^T\mathbf{P} = \mathbf{I}, \widehat{\mathbf{K}}^{(v)} = \sum_{i=1}^{r} \alpha_i \mathbf{K}_i^{(v)}, \\ &\quad \sum_{i=1}^{r} \sqrt{\alpha_i} = 1, \alpha_i \geq 0. \end{aligned} \tag{25}$$

Similar with Eq. (10), w_v in Eq. (26) can be obtained by

$$w_v = 1 \Big/ \left(2\sqrt{\mathrm{Tr}\left(\widehat{\mathbf{K}}^{(v)} - 2\widehat{\mathbf{K}}^{(v)}\mathbf{S} + \mathbf{S}^T\widehat{\mathbf{K}}^{(v)}\mathbf{S}\right)}\right). \tag{26}$$

4.1. Optimization algorithm of MVCMK

Here we present an efficient updating algorithm to optimize the problem in Eq. (26). Specifically, we will iteratively update **S**, **P**, w_v and α until the procedure convergence.

(1) Optimizing Eq. (26) w.r.t. **S**, **P** and w_v with fixed α: $\widehat{\mathbf{K}}^{(v)}$ can be directly calculated according to Eq. (25), and the optimization problem is exactly Eq. (15). Then we only need to use Algorithm 2 with $\widehat{\mathbf{K}}^{(v)}$ as the input kernel matrix.

(2) Optimizing Eq. (26) w.r.t. α with fixed **S**, **P** and w_v: It is clear that Eq. (26) can be rewritten as

$$\begin{aligned} &\min_{\alpha} \sum_{i=1}^{r} \alpha_i q_i \\ &s.t. \quad \sum_{i=1}^{r} \sqrt{\alpha_i} = 1, \alpha_i \geq 0, \end{aligned} \tag{27}$$

Table 3
Clustering performance on Cornell (%).

Method	ACC	Purity	NMI
SC(1)	35.08(1.65)	45.95(2.34)	12.04(2.04)
SC(2)	38.82(1.11)	51.59(0.28)	15.95(0.90)
Co-train	39.90(2.22)	46.56(0.67)	14.55(1.09)
Co-reg	42.56(0.73)	44.21(0.23)	9.88(0.44)
MVKKM	37.59(3.77)	44.51(0.84)	11.19(2.19)
RMKMC	42.77(1.39)	44.72(0.67)	9.63(4.42)
AMGL	42.87(0.28)	43.90(0.28)	8.92(0.25)
MLAN	42.56(0.00)	55.38(0.00)	24.69(0.00)
MVCSK	**50.26(0.00)**	**62.31(0.00)**	**29.52(0.00)**
MVCMK	**62.05(0.00)**	**71.79(0.00)**	**37.96(0.00)**

Table 4
Clustering performance on Washington (%).

Method	ACC	Purity	NMI
SC(1)	54.09(2.83)	62.96(0.94)	19.97(1.10)
SC(2)	51.04(0.39)	56.96(0.00)	16.52(0.05)
Co-train	54.87(3.38)	63.48(0.53)	19.75(1.52)
Co-reg	56.96(3.46)	59.57(3.97)	17.49(2.92)
MVKKM	48.78(1.78)	50.00(2.30)	9.50(5.01)
RMKMC	56.87(9.26)	58.26(9.55)	16.39(11.21)
AMGL	47.39(0.00)	48.26(0.00)	13.79(0.00)
MLAN	62.55(0.00)	71.74(0.00)	**39.62(0.00)**
MVCSK	**65.65(0.00)**	**86.96(0.00)**	27.07(0.00)
MVCMK	**72.17(0.00)**	**84.78(0.00)**	**43.18(0.00)**

Table 5
Clustering performance on Winsconsin (%).

Method	ACC	Purity	NMI
SC(1)	42.49(0.21)	52.45(0.00)	8.56(0.12)
SC(2)	49.81(1.13)	52.00(0.49)	9.05(1.26)
Co-train	42.19(1.91)	51.70(2.15)	8.04(1.02)
Co-reg	47.25(0.32)	47.77(0.21)	9.18(0.52)
MVKKM	49.75(2.99)	53.08(1.90)	11.35(3.11)
RMKMC	52.55(4.61)	54.66(2.62)	12.45(1.53)
AMGL	47.09(0.17)	47.55(0.00)	9.10(0.35)
MLAN	49.81(0.00)	65.66(0.00)	**31.03(0.00)**
MVCSK	**64.91(0.00)**	**86.79(0.00)**	23.06(0.00)
MVCMK	**68.68(0.00)**	**75.85(0.00)**	**39.74(0.00)**

Table 6
Clustering performance on BBCSport (%).

Method	ACC	Purity	NMI
SC(1)	34.31(2.82)	36.55(2.08)	13.81(4.02)
SC(2)	39.48(4.37)	40.86(4.42)	15.88(5.52)
SC(3)	34.31(4.89)	36.90(4.24)	9.62(4.94)
SC(4)	35.52(2.68)	37.41(2.63)	13.46(3.53)
Co-train	31.90(0.61)	33.45(0.72)	7.75(1.22)
Co-reg	32.41(1.16)	34.83(1.79)	6.54(1.26)
MVKKM	37.24(6.38)	38.97(5.80)	14.47(5.69)
RMKMC	34.48(3.90)	36.72(4.29)	10.11(3.29)
AMGL	43.28(3.42)	45.52(4.28)	28.83(1.89)
MLAN	38.45(0.00)	48.79(0.00)	33.63(0.00)
MVCSK	**46.55(0.00)**	**51.28(0.00)**	**39.56(0.00)**
MVCMK	**45.69(0.00)**	**52.24(0.00)**	**37.09(0.00)**

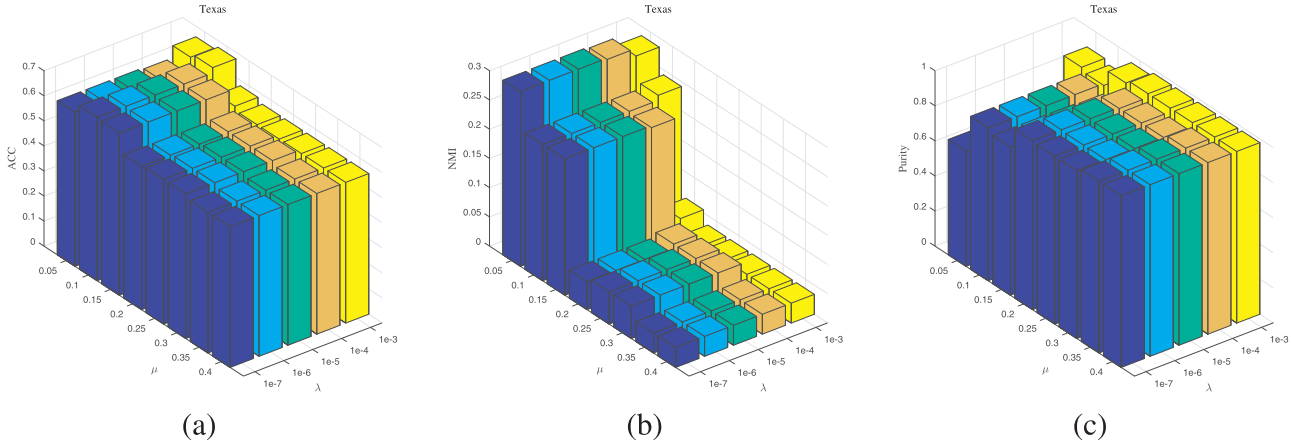


Fig. 1. Clustering performance w.r.t. different parameter settings on Texas.

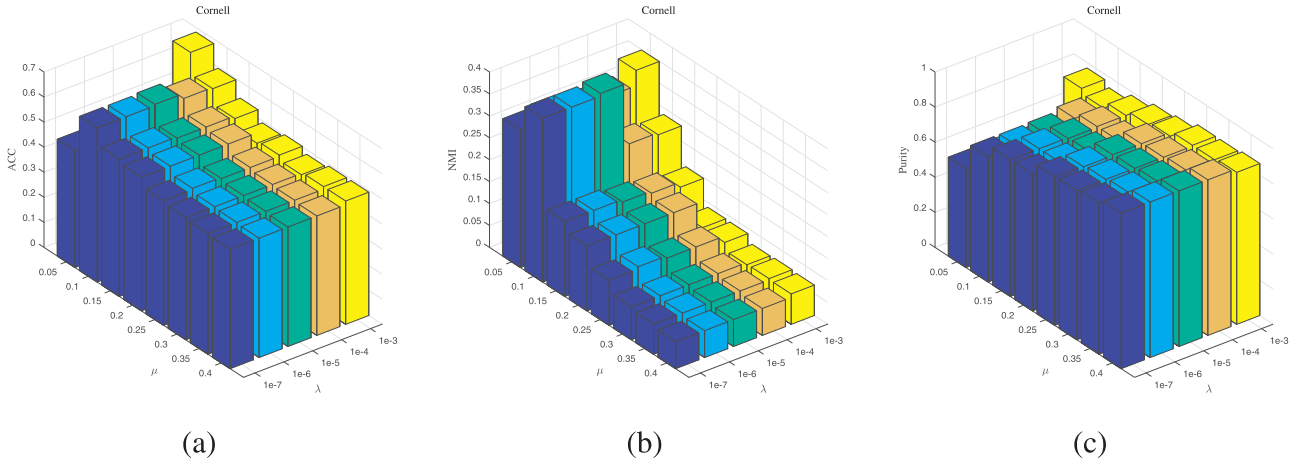


Fig. 2. Clustering performance w.r.t. different parameter settings on Cornell.

Table 7
Clustering performance on BBC (%).

Method	ACC	Purity	NMI
SC(1)	26.03(0.00)	26.34(0.00)	11.11(0.00)
SC(2)	26.18(0.00)	26.50(0.00)	11.21(0.00)
SC(3)	25.63(0.00)	26.15(0.07)	11.39(0.03)
Co-train	25.85(0.04)	26.01(0.04)	16.50(0.07)
Co-reg	25.24(0.37)	26.23(0.11)	13.50(0.04)
MVKKM	37.73(8.60)	37.87(8.58)	16.41(1.64)
RMKMC	25.91(0.28)	26.10(0.16)	10.87(0.10)
AMGL	25.80(0.09)	26.21(0.13)	10.97(0.15)
MLAN	47.59(0.00)	47.59(0.00)	28.86(0.00)
MVCSK	**60.96(0.00)**	**72.32(0.00)**	**35.26(0.00)**
MVCMK	**69.87(0.00)**	**70.74(0.00)**	**49.65(0.00)**

where

$$q_i = w_v \mathrm{Tr}\left(\widehat{\mathbf{K}}^{(v)} - 2\widehat{\mathbf{K}}^{(v)}\mathbf{S} + \mathbf{S}^T\widehat{\mathbf{K}}^{(v)}\mathbf{S}\right). \tag{28}$$

Note that the Lagrange function of Eq. (29) is

$$J(\alpha) = \alpha^T\mathbf{q} + \xi(1 - \textstyle\sum_{i=1}^{r}\sqrt{\alpha_i}), \tag{29}$$

where ξ is the Lagrangian multiplier. Using the KKT condition [41] with $\frac{\partial J(\alpha)}{\partial \alpha_i} = 0$ and the constraint $\sum_{i=1}^{r}\sqrt{\alpha_i} = 1$, then we have

$$\alpha_i = \left(q_i \textstyle\sum_{j=1}^{r}\frac{1}{q_j}\right)^{-2}. \tag{30}$$

The details of the proposed MVCMK optimization are given in Algorithm 1.

Table 8
Clustering performance on NUS (%).

Method	ACC	Purity	NMI
SC(1)	19.65(0.70)	36.42(0.35)	11.39(0.08)
SC(2)	17.29(0.64)	35.16(0.15)	11.02(0.14)
SC(3)	18.13(0.11)	36.74(0.06)	12.48(0.12)
SC(4)	15.29(0.18)	34.53(0.18)	8.93(0.05)
SC(5)	15.21(0.17)	32.08(0.14)	7.12(0.13)
Co-train	19.93(0.37)	36.20(0.13)	11.89(0.12)
Co-reg	19.09(0.22)	36.05(0.35)	11.56(0.30)
MVKKM	17.05(0.71)	32.43(0.87)	9.27(0.36)
RMKMC	19.19(0.29)	37.18(0.51)	13.69(0.50)
AMGL	24.59(0.03)	**38.03(0.08)**	**16.84(0.05)**
MLAN	**29.84(0.00)**	30.62(0.00)	3.29(0.00)
MVCSK	26.20(0.00)	35.54(0.00)	11.25(0.00)
MVCMK	**30.99(0.00)**	**44.51(0.00)**	**14.10(0.00)**

Table 9
Running time of all the multi-view clustering methods (seconds).

Method	Texas	Cornell	Washington	Winconsin	BBCSport	BBC
Co-train	1.37	1.63	1.78	2.37	2.36	140.00
Co-reg	3.77	4.90	5.00	4.60	19.14	691.33
MVKKM	0.03	0.03	0.04	0.04	0.03	4.67
RMKMC	0.85	1.16	1.17	1.38	0.78	11.90
AMGL	0.15	0.17	0.21	0.27	0.08	25.48
MLAN	0.38	0.41	0.51	0.62	0.19	36.89
MVCSK	0.82	1.09	1.86	2.10	0.79	40.54
MVCMK	1.08	1.96	2.83	3.86	1.20	127.93

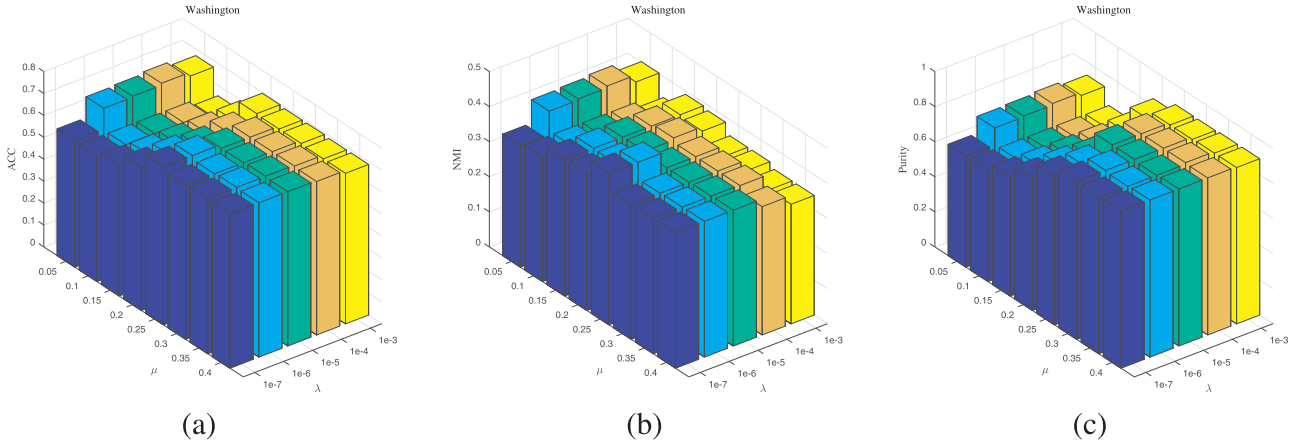


Fig. 3. Clustering performance w.r.t. different parameter settings on Washington.

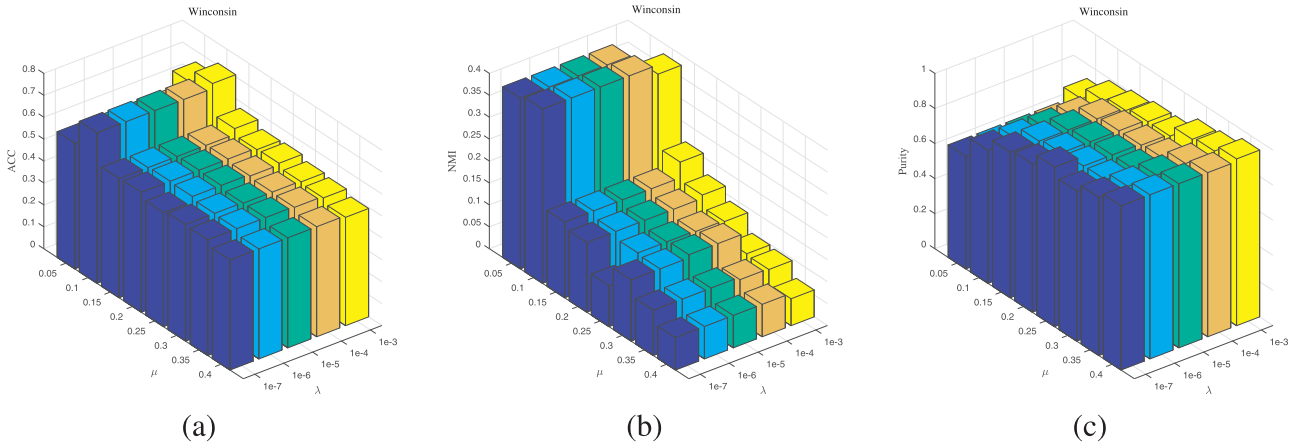


Fig. 4. Clustering performance w.r.t. different parameter settings on Winconsin.

4.2. Time complexity analysis

Here we give a discussion about the computational cost of MVCSK and MVCMK. From Algorithms 2 and1, the time complexity of both MVCSK and MVCMK is $O(n^3)$, where n denotes the number of instances. Besides the iterative updates, our model also needs $O(n^2k)$ to construct the kernel, where $k = \sum_v k^{(v)}$ and $k^{(v)}$ is the number of features of the vth view. In reality, we have $k \ll n$. Thus the overall cost for MVCSK and MVCMK is $O(n^3)$.

5. Experiments

To measure the effectiveness of our MVCSK and MVCMK, we compare them with the following state-of-the-art multi-view clustering methods:

- Co-trained multi-view spectral clustering (Co-train) [21].
- Co-regularized multi-view spectral clustering (Co-reg) [22].
- Multi-view kernel K-means clustering (MVKKM) [28].
- Robust multi-view K-means clustering (RMKMC) [29].
- Multi-view clustering via self-paced learning (MSPL) [42].
- Multi-view clustering via multiple graph learning (AMGL) [36].
- Multi-view clustering with adaptive neighbours (MLAN) [43].

The classic graph-based algorithm, spectral clustering (SC), is also included as baseline. We apply SC on every dataset using each view of features (e.g., SC(1) means performing SC on the 1st view).

5.1. Data sets

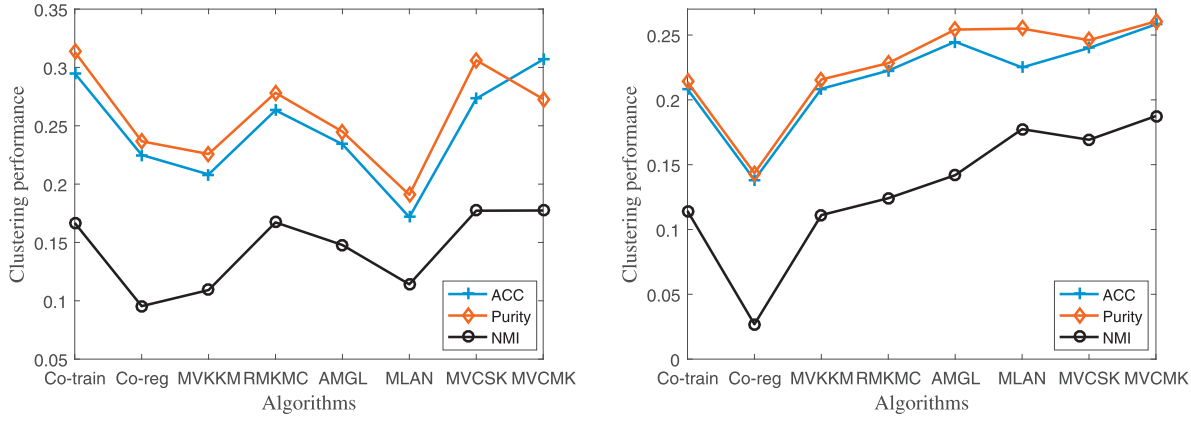
Several benchmark datasets are used to assess the performance of the proposed methods.

The WebKB dataset has been widely used in multi-view learning [44], which contains webpages collected from four universities: Cornell, Texas, Washington and Wisconsin. The webpages are distributed over five classes: student, project, course, staff and faculty and described by two views: the content view and the citation view. Each webpage is denoted by 1703 words in the content view, and the number of citation links between other pages in the citation view.

The datasets BBC and BBCSport are derived from the bbc and bbcsport news corpora which have been previously used in document clustering tasks [45]. The original bbc corpus contains a total of 2225 documents with 5 annotated topic labels, while the original bbcsport corpus contains a total of 737 documents also with 5 annotated labels. From each corpus we constructed new synthetic datasets with 2–4 views. In our experiments, we construct a subset of BBCSport with 5470, 5549 and 5483 words in each view, and a subset of BBC with 1991, 2063, 2113 and 2158 words in each view, respectively.

NUS-WIDE-Object (NUS) [46] is a dataset for object recognition which consists of 30,000 images in 31 classes. In our experiments, we select a subset which contains 11,576 images of 15 classes: bear, birds, boats, book, cars, cat, computer, coral, cow, dog, elk, fish, flags, flowers and fox. Each image is represented by five type of features: color histogram, color moments, color correlation, edge

Fig. 5. The constructed subset of NUS with noise images.



(a) Results on constructed dataset without noise. (b) Results on constructed dataset with noise.

Fig. 6. Clustering performance on the constructed dataset.

distribution and wavelet texture. The specific characteristics of the datasets are given in Table 1.

5.2. Experiment setup

To demonstrate the effectiveness of multiple kernel learning, 9 kernels including linear, nonlinear and polynomial kernels are constructed. Specifically, we design a linear kernel (i.e., $\mathbf{K}(\mathbf{x}, \mathbf{y}) = \mathbf{x}^T\mathbf{y}$), seven Gaussian kernels (i.e., $\mathbf{K}(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x} - \mathbf{y}\|_2^2/h\ell^2)$, where h is the maximal distance between data points and ℓ is searched in {0.01, 0.05, 0.1, 1, 10, 50, 100}), and a polynomial kernel (i.e., $\mathbf{K}(\mathbf{x}, \mathbf{y}) = (1 + \mathbf{x}^T\mathbf{y})^2$).

For other compared methods, the source code is downloaded from the authors' website, and we also follow their experimental setting for fair comparison. We repeat clustering 10 times, and the mean results are recorded.

5.3. Clustering results

We use normalized mutual information (NMI), Purity, and accuracy (ACC) to evaluate the clustering results. These metrics are widely used metrics for clustering [47]. And they are generally positive correlated, i.e., a larger value means a better performance. We highlight the top two mean and standard deviation of the results in boldface.

The clustering results of all datasets are shown in Tables 2–8. We can see the multi-view clustering methods are generally better than the baseline methods on each view, which suggests that integrating the information provided by all views can effectively improve clustering performance. It can be seen that utilizing the multi-view clustering formulations, the hidden structures of data are fully explored. Furthermore, the standard deviations of our methods and MLAN are always 0.00. The reason is that with the Laplacian matrix rank constraint introduced in our model (as described in Theorem 1), once we obtain the target graph, the cluster label of each data point can be directly assigned without any postprocessing such as K-means, which makes the algorithm relatively stable. Furthermore, the weight for each graph is automatically assigned, thus makes the algorithm more deterministic. As a result, the clustering results obtained in each time are almost the same under the same parameter setting, thus the standard deviation approaches to zero. Note that this also shows the robustness of MVCSK and MVCMK.

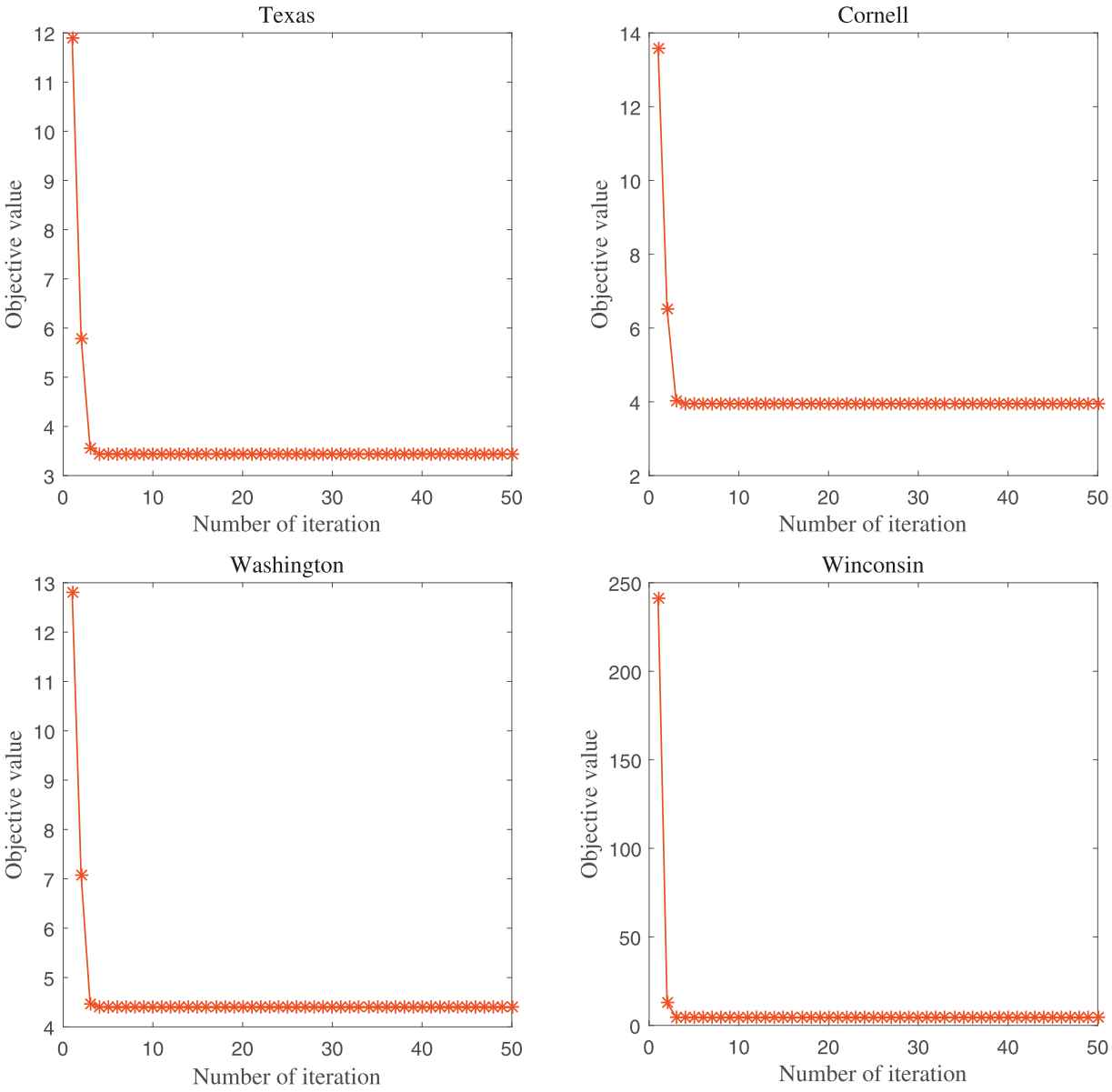


Fig. 7. Convergence curve of MVCMK on four datasets.

The clustering results of the proposed methods are generally better than that of other compared methods, which shows the effectiveness of our methods. The superiority of our methods arises the aspect that the multi-view clustering performance can be effectively improved by leveraging the interactions of constructing the most accurate similarity graph, allocating ideal weight for each view automatically and finding the cluster indicator matrix in a joint framework. Moreover, MVCMK outperforms MVCSK on all datasets in most cases, which demonstrates that the final results can be further improved by utilizing the multiple kernel learning scheme.

5.4. Parameter tuning

Here we report the clustering performance under different parameter settings. The sensitivity of our model is investigated w.r.t the parameter μ and the regularization parameter λ.

From Figs. 1–4, it is obvious that our model is quite insensitive to μ and λ in terms of ACC and Purity. On the other hand, our method is a little sensitive to μ and λ in terms of NMI on some datasets.

5.5. Real world data sets with noises

Clustering methods are usually sensitive to outliers and noisy data, which greatly impair the final results in practical applications. To further illustrate the effectiveness of the proposed methods, we perform our methods on a dataset with outliers. Hence, we construct a new dataset by randomly sampling 120 images for each class from NUS. We select ten classes and thus a new dataset contains 1200 images is constructed. For the new dataset, the additional noises ("salt and pepper" noise with thirty percent noise density [48]) is added to the first 20 images of each class. As an example, the constructed subset of the dataset with noise images is shown in Fig. 5. Owing to the limited space, we only show 12 images of each class.

The clustering performance on the constructed dataset with noise of all compared methods is plotted in Fig. 6. For fair compar-

ison, the clustering performance on the constructed dataset without noise is also recorded. we can see that the clustering performance obtained by the compared methods as shown in Fig. 6 (a) is generally better than that in Fig. 6 (b). This is due to the fact that the clustering performance is affected by the noise. On the other hand, it can be seen that the performance of our method is relatively stable compared with other methods. Furthermore, the clustering results of MVCSK and MVCMK are generally better than that of other compared methods. And MVCMK consistently outperforms these methods, which further validates the effectiveness of handling noises.

5.6. Computational performance

The computational speed of the proposed method as well as other compared methods is recorded. We perform our experiment with Matlab R2016a on a machine with Core 12 Duad 2.6 GHz and 64 GB memory. The results of all methods are.

As shown in Table 9, the execution time of the graph based methods is generally higher than that of other methods due to the extra construction of the graph similarity. Furthermore, the execution time of Co-reg is the highest among the eight multi-view clustering methods, while the execution time of MVKKM is the lowest among the eight methods. Generally speaking, our methods are faster than Co-reg, slower than MVKKM and RMKMC but in line with other methods on most datasets, which illustrates the effectiveness of our methods.

5.7. Convergence study

We also empirically show how fast our method will converge. Fig. 7 shows the convergence curves of MVCMK on all the datasets, where *x*-axis denotes the number of iterations, while y-axis denotes the objective value.

We can see that the updating rules for MVCMK converge very fast, usually within 10 iterations.

6. Conclusion

In this paper, we present a novel multi-view learning model which simultaneously performs multi-view clustering task and learns similarity relationships in kernel spaces. The obtained optimal graph can be directly partitioned into exact c connected components if there are c clusters. Furthermore, the proposed model can automatically assign optimal weight for each view without additional parameters as previous methods do. Since the performance is often sensitive to the input kernel matrix, we further extend our model with a multiple kernel learning ability. With this joint model, we can simultaneously accomplish three subtasks of constructing the most accurate similarity graph, allocating ideal weight for each view automatically and finding the cluster indicator matrix. By this joint learning, each subtask can be mutually enhanced. Extensive experimental results on several benchmark datasets demonstrate that the proposed model outperforms other state-of-the-art multi-view clustering algorithms. In our future work, we are interested in applying the proposed framework to other machine learning areas such as semi-supervised learning and classification problems.

Acknowledgments

This work was partially supported by NSF China (No.61572111), Startup funding of UESTC (Nos.A1098531023601041 and G05QNQR004), two Fundamental Research Fund for the Central Universities of China (Nos.ZYGX2016Z003 and ZYGX2017KYQD177), ARC Future Fellowship FT130100746, ARC grant LP150100671 and DP180100106.

References

[1] C. Peng, Z. Kang, S. Cai, Q. Cheng, Integrate and conquer: double-sided two-dimensional k-means via integrating of projection and manifold construction, ACM Trans. Intell. Syst. Technol. 9 (5) (2018) 57–68.
[2] Z. Kang, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, Knowl. Based Syst. (2018). in press, https://doi.org/10.1016/j.knosys.2018.09.009.
[3] S. Huang, P. Zhao, Y. Ren, T. Li, Z. Xu, Self-paced and soft-weighted nonnegative matrix factorization for data representation, Knowl. Based Syst. (2018). in press, https://doi.org/10.1016/j.knosys.2018.10.003.
[4] S. Huang, H. Wang, D. Li, Y. Yang, T. Li, Spectral co-clustering ensemble, Knowl. Based Syst. 84 (2015) 46–55.
[5] F. Nie, X. Wang, H. Huang, Clustering and projected clustering with adaptive neighbors, in: Proceedings of the Twentieth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2014, pp. 977–986.
[6] S. Huang, H. Wang, T. Li, Y. Yang, T. Li, Constraint co-projections for semi-supervised co-clustering, IEEE Trans. Cybern. 46 (12) (2016) 3047–3058.
[7] M.B. Blaschko, C.H. Lampert, Correlational spectral clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2008, pp. 1–8.
[8] S. Hou, L. Chen, D. Tao, S. Zhou, W. Liu, Y. Zheng, Multi-layer multi-view topic model for classifying advertising video, Pattern Recognit. 68 (2017) 66–81.
[9] J. Yu, D. Tao, Y. Rui, J. Cheng, Pairwise constraints based multiview features fusion for scene classification, Pattern Recognit. 46 (2) (2013) 483–496.
[10] L. Zhang, Q. Zhang, L. Zhang, D. Tao, X. Huang, B. Du, Ensemble manifold regularized sparse low-rank approximation for multiview feature embedding, Pattern Recognit. 48 (10) (2015) 3102–3112.
[11] S. Huang, Y. Ren, Z. Xu, Robust multi-view data clustering with multi-view capped-norm k-means, Neurocomputing 311 (2018) 197–208.
[12] S. Huang, Z. Xu, J. Lv, Adaptive local structure learning for document co-clustering, Knowl. Based Syst. 148 (2018) 74–84.
[13] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, Knowl. Based Syst. 158 (2018) 1–8.
[14] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, W. Gao, Late fusion incomplete multi-view clustering, IEEE Trans. Pattern Anal. Mach. Intell. (2018) in press, doi:10.1109/TPAMI.2018.2879108.
[15] M. White, X. Zhang, D. Schuurmans, Y.-l. Yu, Convex multi-view subspace learning, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1673–1681.
[16] 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.
[17] Y. Guo, Convex subspace representation learning from multi-view data., in: Proceedings of the Twenty-Seventh AAAI Conference on Artificial Intelligence, 2013, pp. 387–393.
[18] Q. Yin, S. Wu, R. He, L. Wang, Multi-view clustering via pairwise sparse subspace representation, Neurocomputing 156 (2015) 12–21.
[19] X. Zhang, X. Zhang, H. Liu, Multi-task multi-view clustering for non-negative data, in: Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence, 2015, pp. 4055–4061.
[20] 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.
[21] A. Kumar, H. Daumé, A co-training approach for multi-view spectral clustering, in: Proceedings of the Twenty-Eighth International Conference on Machine Learning, 2011, pp. 393–400.
[22] A. Kumar, P. Rai, H. Daume, Co-regularized multi-view spectral clustering, in: Proceedings of the Advances in Neural Information Processing Systems, 2012, pp. 1413–1421.
[23] Y. Zhao, Y. Dou, X. Liu, T. Li, A novel multi-view clustering method via low-rank and matrix-induced regularization, Neurocomputing 216 (2016) 342–350.
[24] F.d.A. de Carvalho, F.M. de Melo, Y. Lechevallier, A multi-view relational fuzzy c-medoid vectors clustering algorithm, Neurocomputing 163 (2015) 115–123.
[25] Z. Kang, C. Peng, Q. Cheng, Kernel-driven similarity learning, Neurocomputing 267 (2017) 210–219.
[26] 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.
[27] Y.-M. Xu, C.-D. Wang, J.-H. Lai, Weighted multi-view clustering with feature selection, Pattern Recognit. 53 (2016) 25–35.
[28] G. Tzortzis, A. Likas, Kernel-based weighted multi-view clustering, in: Proceedings of the Twelfth IEEE International Conference on Data Mining, 2012, pp. 675–684.
[29] X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2013, pp. 2598–2604.
[30] C.-D. Wang, J.-H. Lai, S.Y. Philip, Multi-view clustering based on belief propagation, IEEE Trans. Knowl. Data Eng. 28 (4) (2016) 1007–1021.
[31] L. Zong, X. Zhang, L. Zhao, H. Yu, Q. Zhao, Multi-view clustering via multi-manifold regularized non-negative matrix factorization, Neural Netw. 88 (2017) 74–89.
[32] E. Elhamifar, R. Vidal, Sparse subspace clustering, in: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2009, pp. 2790–2797.
[33] Z. Kang, C. Peng, Q. Cheng, Twin learning for similarity and clustering: a unified kernel approach, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2080–2086.

[34] C. Zhang, F. Nie, S. Xiang, A general kernelization framework for learning algorithms based on kernel PCA, Neurocomputing 73 (4–6) (2010) 959–967.
[35] I. Daubechies, R. DeVore, M. Fornasier, C.S. Güntürk, Iteratively reweighted least squares minimization for sparse recovery, Commun. Pure Appl. Math. 63 (1) (2010) 1–38.
[36] F. Nie, J. Li, X. Li, Parameter-free auto-weighted multiple graph learning: a framework for multiview clustering and semi-supervised classification, in: Proceedings of the International Joint Conference on Artificial Intelligence, 2016, pp. 1881–1887.
[37] B. Mohar, The Laplacian spectrum of graphs, in: Graph Theory, Combinatorics, and Applications, 1991, pp. 871–898.
[38] K. Fan, On a theorem of Weyl concerning eigenvalues of linear transformations i, Proc. Natl. Acad. Sci. 35 (11) (1949) 652–655.
[39] F. Nie, H. Huang, X. Cai, C.H. Ding, Efficient and robust feature selection via joint $\ell_{2,1}$-norms minimization, in: Proceedings of the NIPS, 2010, pp. 1813–1821.
[40] H. Zeng, Y.-M. Cheung, Feature selection and kernel learning for local learning-based clustering, IEEE Trans. Pattern Anal. Mach. Intell. 33 (8) (2011) 1532–1547.
[41] S. Boyd, L. Vandenberghe, Convex Optimization, Cambridge University Press, 2004.
[42] C. Xu, D. Tao, C. Xu, Multi-view self-paced learning for clustering, in: Proceedings of the International Conference on Artificial Intelligence, 2015, pp. 3974–3980.
[43] F. Nie, C. Guohao, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: Proceedings of the AAAI Conference on Artificial Intelligence, 2017, pp. 2408–2414.
[44] G. Bisson, C. Grimal, Co-clustering of multi-view datasets: a parallelizable approach, in: Proceedings of the IEEE International Conference on Data Mining, 2013, pp. 828–833.
[45] D. Greene, P. Cunningham, A matrix factorization approach for integrating multiple data views, in: Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases, 2009, pp. 423–438.
[46] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, Y. Zheng, NUS-WIDE: a real-world web image database from national university of Singapore, in: Proceedings of the ACM International Conference on Image and Video Retrieval, 2009.
[47] S. Huang, H. Wang, T. Li, T. Li, Z. Xu, Robust graph regularized nonnegative matrix factorization for clustering, Data Min. Knowl. Discov. 32 (2) (2018) 483–503.
[48] P. Rosin, J. Collomosse, Image and Video-Based Artistic Stylisation, Springer London, 2013.

Shudong Huang received the M.Sc. degree in the School of Information Science and Technology from Southwest Jiaotong University, China, in 2015. He is currently a Ph.D. student in the School of Computer Science and Engineering, University of Electronic Science and Technology of China, China. His research interests are machine learning, data mining, semi-supervised learning and ensemble learning. He is a student member of CCF.

Zhao Kang received his Ph.D. degree from the Department of Computer Science at Southern Illinois University in May 2017. At present, Dr. Kang Zhao focuses on the research of theory and method design in machine learning , as well as applying those methods on practical problems in computer vision, social network, information retrieval and data mining. He published more than 20 papers in top conferences and journals in the area of artificial intelligence and data mining, including AAAIDEPRGKDDDMMKMM TISTM TKDDeurocomputing. At the same time, he has been invited to serve as reviewer and member of the Procedural Committee for the top journals and conferences in related fields for times.

Ivor W. Tsang is an ARC Future Fellow and Professor of Artificial Intelligence, at University of Technology Sydney (UTS). He is also the Research Director of the UTS Flagship Research Centre for Artificial Intelligence (CAI). His research focuses on transfer learning, feature selection, crowd intelligence, big data analytics for data with extremely high dimensions in features, samples and labels, and their applications to computer vision and pattern recognition. He has more than 160 research papers published in top-tier journal and conference papers. According to Google Scholar, his H-index is 47. In 2009, Prof Tsang was conferred the 2008 Natural Science Award (Class II) by Ministry of Education, China, which recognized his contributions to kernel methods. In 2013, Prof Tsang received his prestigious Australian Research Council Future Fellowship for his research regarding Machine Learning on Big Data. In addition, he had received the prestigious IEEE Transactions on Neural Networks Outstanding 2004 Paper Award in 2007, the 2014 IEEE Transactions on Multimedia Prize Paper Award, and a number of best paper awards and honors from reputable international conferences, including the Best Student Paper Award at CVPR 2010, and the Best Paper Award at ICTAI 2011.

Zenglin Xu received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong. He is currently a full professor in University of Electronic Science & Technology of China. He has been working at Michigan State University, Cluster of Excellence at Saarland University and Max Planck Institute for Informatics, and later Purdue University. Dr. Xu's research interests include machine learning and its applications in information retrieval, health informatics, and social network analysis. He has been elected in the2013's China Youth 1000-talent Program. He is the recipient of the outstanding student paper honorable mention of AAAI 2015, the best student paper runner up of ACML 2016, and the 2016 young researcher award from APNNS