



## Clustering with similarity preserving

Zhao Kang<sup>a</sup>, Honghui Xu<sup>a</sup>, Boyu Wang<sup>b</sup>, Hongyuan Zhu<sup>c</sup>, Zenglin Xu<sup>a,\*</sup>

<sup>a</sup>SMILE Lab, School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China

<sup>b</sup>Department of Computer Science, University of Western Ontario, Canada

<sup>c</sup>Institute for Infocomm Research, A\*Star, Singapore

### ARTICLE INFO

#### Article history:

Received 6 October 2018

Revised 16 May 2019

Accepted 26 July 2019

Available online 31 July 2019

Communicated by Dr. Zhu Jianke

#### Keywords:

Clustering

Kernel method

Similarity preserving

Multiple kernel learning

Graph learning

### ABSTRACT

Graph-based clustering has shown promising performance in many tasks. A key step of graph-based approach is the similarity graph construction. In general, learning graph in kernel space can enhance clustering accuracy due to the incorporation of nonlinearity. However, most existing kernel-based graph learning mechanisms is not similarity-preserving, hence leads to sub-optimal performance. To overcome this drawback, we propose a more discriminative graph learning method which can preserve the pairwise similarities between samples in an adaptive manner for the first time. Specifically, we require the learned graph be close to a kernel matrix, which serves as a measure of similarity in raw data. Moreover, the structure is adaptively tuned so that the number of connected components of the graph is exactly equal to the number of clusters. Finally, our method unifies clustering and graph learning which can directly obtain cluster indicators from the graph itself without performing further clustering step. The effectiveness of this approach is examined on both single and multiple kernel learning scenarios in several datasets.

© 2019 Elsevier B.V. All rights reserved.

### 1. Introduction

Discovering clusters in unlabeled data is one of the most fundamental scientific tasks, with an endless list of practical applications in data mining, pattern recognition, and machine learning [1–5]. It is well-known that labels are expensive to obtain, so clustering techniques are useful tools to process data and to reveal its underlying structure.

Over the past decades, a number of clustering techniques have been developed [6–9]. One main class of clustering methods is K-means and its various extensions. To some extent, these techniques are distance-based methods. K-means has been extensively investigated since its introduction in 1957 by Lloyd [10], due to its simplicity and effectiveness. However, it is only suitable for data points that are evenly spread around some centroids [11,12]. To make it work under general circumstances, much effort has been spent on mapping data to a certain space. One representative approach is using kernel method. The first Kernel K-means algorithm was proposed in 1998 [13]. Although some data points cannot be separated in the original data representation, they are linearly separable in kernel space.

Recently, robust Kernel K-means (RKKM) method has been developed [14]. Different from other K-means algorithms, RKKM uses  $\ell_{21}$ -norm to evaluate the fidelity term. Consequently, RKKM can alleviate the adverse effects of noise and outliers considerably. It shows that RKKM can achieve superior performance on a number of real-world data sets. However, its performance still depends on the choice of the kernel function.

Graph-based algorithms, as another main category of clustering methods, have been drawing growing attention. Among them, spectral clustering is a leading and highly popular method due to its ability in incorporating manifold information with good performance [15,16]. In particular, it embeds the data into the eigenspace of the Laplacian matrix, derived from the pairwise similarities between data points [17]. A commonly used way of similarity measure is the Gaussian kernel [18]. Nevertheless, it is challenging to select an appropriate scaling factor  $\sigma$  [19]. Kernel spectral clustering (KSC) [20] and its variants [21] have also been proposed.

Recently, a novel approach which models graph construction as an optimization problem has been proposed [22–26]. It works by either performing adaptive local structure learning or representing each data point as a weighted combination of other data points. The second approach can capture the global structure information and can be easily extended to kernel space [27,28]. That is to say, one seeks to learn a high-quality graph from artificially constructed kernel matrix. These methods are free of similarity

\* Corresponding author.

E-mail addresses: [zkang@uestc.edu.cn](mailto:zkang@uestc.edu.cn), [zenglin@gmail.com](mailto:zenglin@gmail.com) (Z. Xu).

metrics or kernel parameters, thus they are more appealing to real-world applications.

Although the above approach has shown much better performance than traditional methods, it also causes some information loss. In particular, it learns similarity graph from the data itself without considering other prior information. Consequently, some similarity information might get lost, which should be helpful for our graph learning [29,30]. On the other hand, preserving similarity information has been shown to be important for feature selection [31]. In [31], new feature vector  $f$  is obtained by maximizing  $f^T \hat{K} f$ , where  $\hat{K}$  is the refined similarity matrix derived from original kernel matrix  $K$ . In this paper, we propose a way to preserve the similarity information between samples when we learn the graph and cluster labels. To the best of our knowledge, this is the first work that develops similarity preserving strategy for graph learning.

It is necessary to point out that the key point of this paper is the similarity preserving concept. Though there are many similarity learning methods in the literature, they often ignore to explicitly retain structure information of original data. Concretely, we expect our learned similarity matrix  $Z$  approximates pre-defined kernel matrix  $K$  to some extent. The quality of similarity matrix is crucial to many tasks, such as graph embedding [32], where the low-dimensional representation is expected to respect the neighborhood relation characterized by  $Z$ .

In addition, most existing graph-based clustering methods perform clustering in two separate steps [15,24,25,33]. Specifically, they first construct a graph. Then, the obtained graph is inputted to the spectral clustering algorithm. In this approach, the quality of the graph is not guaranteed, which might not be suitable for subsequent clustering [22,34]. In this paper, the structure information of the graph is explicitly considered in our model, so that the component number in the learned graph is equal to the number of clusters. Then, we can directly obtain cluster indicators from the graph itself without performing further graph cut or K-means clustering steps. Extensive experimental results validate the effectiveness of our proposed method.

The contributions of this paper are two-fold:

1. Our proposed model has the capability of similarity preserving. This is the first attempt to preserve the sample's similarity information when we construct the similarity graph. Consequently, the quality of the learned graph would be enhanced.
2. Cluster structure is seamlessly incorporated into our objective function. As a result, the component number in the learned graph is equal to the number of clusters, such that the vertices in each connected component of the graph are partitioned into one cluster. Therefore, we directly obtain cluster indicators from the graph itself without performing further graph cut or K-means clustering steps.

**Notations.** Given a data matrix  $X \in \mathcal{R}^{m \times n}$  with  $m$  features and  $n$  samples, we denote its  $(i, j)$ th element and  $i$ th column as  $x_{ij}$  and  $X_i$ , respectively. The  $\ell_2$ -norm of vector  $x$  is represented by  $\|x\| = \sqrt{x^T \cdot x}$ , where  $x^T$  is the transpose of  $x$ . The squared Frobenius norm is defined as  $\|X\|_F^2 = \sum_{ij} x_{ij}^2$ .  $I$  represents the identity matrix with the proper size.  $Z \geq 0$  means all the elements of  $Z$  are nonnegative.  $\langle \cdot, \cdot \rangle$  denotes the inner product of two matrices.

## 2. Preliminaries

In this section, we give a brief overview of two popular similarity learning techniques which have been developed recently.

### 2.1. Adaptive local structure learning

For each data point  $X_i$ , it can be connected to data point  $X_j$  with probability  $z_{ij}$ . Closer points should have a larger probability, thus  $z_{ij}$  characterizes the similarity between  $X_i$  and  $X_j$  [22,35]. Since  $z_{ij}$  has the negative correlation with the distance between  $X_i$  and  $X_j$ , the determination of  $z_{ij}$  can be achieved by optimizing the following problem:

$$\min_{Z_i} \sum_{j=1}^n (\|X_i - X_j\|^2 z_{ij} + \gamma z_{ij}^2),$$

$$\text{s.t. } Z_i^T \mathbf{1} = 1, 0 \leq z_{ij} \leq 1, \quad (1)$$

where  $\gamma$  is the trade-off parameter. Here,  $Z$  is adaptively learned from the data. This idea has recently been applied in a number of problems. Nonnegative matrix factorization [36,37], feature selection [38], multi-view learning [39], just to name a few. One limitation of this method is that it can only capture the local structure information and thus the performance might be deteriorated.

### 2.2. Adaptive global structure learning

To explore the global structure information, methods based on self-expression, have become increasingly popular in recent years [23,40]. The basic idea is to encode each datum as a weighted combination of other samples, i.e., its direct neighbors and reachable indirect neighbors. If  $X_j$  is quite similar to  $X_i$ , coefficient  $z_{ij}$ , which denotes the contribution from  $X_j$  to  $X_i$ , should be large. From this point of view,  $z_{ij}$  can be viewed as the similarity between the data points. The corresponding optimization problem can be formulated as:

$$\min_Z \frac{1}{2} \|X - XZ\|_F^2 + \gamma \|Z\|_F^2 \text{ s.t. } Z \geq 0 \quad (2)$$

This has drawn significant attention and achieved impressive performance in a number of applications, including face recognition [41], motion segmentation [24,25], semi-supervised learning [42].

As a matter of fact, Eq. (2) is related to some dimension reduction methods. For example, in Locally Linear Embedding (LLE),  $k$ -nearest neighbors are first identified for each data point [43]. Then each data point is reconstructed by a linear combination of its  $k$  nearest neighbors. By contrast, Eq. (2) uses all data points and determines the neighbors automatically according to the optimization result. Thus, it is supposed to capture the global structure information. Eq. (2) is different from Locality Preserving Projections (LPP), which tries to preserve the neighborhood structure during the dimension reduction process [44]. LPP uses a predefined similarity matrix to characterize the neighbor relations, while Eq. (2) is trying to learn this similarity matrix automatically from data. For Laplacian Eigenmaps (LE), a similarity graph matrix is also predefined [45]. On the other hand, Principal Component Analysis (PCA) aims to find a projection so that the variance is maximized in low-dimensional space, which is less relevant to the similarity learning methods.

To capture the nonlinear structure information of data, Eq. (2) can be easily extended to kernel space, which gives

$$\min_Z \frac{1}{2} \text{Tr}(K - 2KZ + Z^T K Z) + \gamma \|Z\|_F^2 \text{ s.t. } Z \geq 0, \quad (3)$$

where  $\text{Tr}(\cdot)$  is the trace operator and  $K$  is the kernel matrix of  $X$ . This model recovers the linear relations among the data in the new representation, and thus the nonlinear relations in the original space. Eq. (3) is more general than Eq. (2) and reduces to Eq. (2) if a linear kernel function is applied.

### 3. Similarity preserving clustering

The aforementioned two learning mechanisms lead to much better performance than traditional similarity measure based techniques in many real-world applications. However, they ignore some important information. Specifically, as they operate on the data itself, some data relation information might get lost [29]. Since we seek to learn a high-quality similarity graph, data relation information would be crucial to our task. In this paper, we aim to retain this information.

Because the kernel matrix  $K$  itself contains similarity information of data points, we expect  $Z$  to be close to  $K$ . To this end, we optimize the following objective function

$$\max_Z \langle K, Z \rangle \Leftrightarrow \max_Z \text{Tr}(KZ) \Leftrightarrow \min_Z -\text{Tr}(KZ). \quad (4)$$

Although we claim similarity preserving, Eq. (4) also keeps dissimilarity information. For example, if points  $i$  and  $j$  are from different clusters,  $K_{ij} = K_{ji} = 0$ , then  $Z_{ij} = Z_{ji} = 0$  would hold. Note that we already have  $-\text{Tr}(KZ)$  term in Eq. (3). Hence we can combine Eq. (4) and Eq. (3) by introducing a coefficient  $\alpha > 1$ , we have

$$\min_Z \frac{1}{2} \text{Tr}(K + Z^T K Z) - \alpha \text{Tr}(KZ) + \gamma \|Z\|_F^2 \text{ s.t. } Z \geq 0. \quad (5)$$

Although we just make a small modification to Eq. (3), it makes a lot of sense in practice. By tuning parameter  $\alpha$ , we can control how much relation information we want to keep from the original kernel matrix. In particular,  $\alpha$  can avoid the conflicts between the pre-computed similarity  $K$  and the learned similarity  $Z$ . If  $K$  is not suitable to reveal the underlying relationships among samples, we just set  $\alpha = 1$ , which means that there is no similarity preserving effect. The influence of selecting parameter  $\alpha$  is elaborated in Section 5.2.4.

Eq. (5) provides a framework to learn graph matrix  $Z$  with similarity preservation. Further clustering is achieved by using spectral clustering and K-means clustering on the learned graph. These separate steps often lead to suboptimal solutions [22] and K-means is sensitive to the initialization of cluster centers. To this end, we propose to unify clustering with graph learning, so that two tasks can be simultaneously achieved. Specifically, if there are  $c$  clusters in the data, we hope to learn a graph with exactly  $c$  number of connected components. Obviously, Eq. (5) can hardly satisfy such a constraint. To this end, we leverage the following theorem:

**Theorem 1** [46]. *The number of connected components  $c$  is equal to the multiplicity of zero as an eigenvalue of its Laplacian matrix  $L$ .*

Since  $L$  is positive semi-definite, it has  $n$  non-negative eigenvalues  $\lambda_n \geq \dots \geq \lambda_2 \geq \lambda_1 \geq 0$ . Theorem 1 indicates that if  $\sum_{i=1}^c \lambda_i = 0$  is satisfied, the graph  $Z$  would be ideal and the data points are already clustered into  $c$  clusters. According to Fan's theorem [47], we obtain

$$\sum_{i=1}^c \lambda_i = \min_{F \in \mathbb{R}^{n \times c}, F^T F = I} \text{Tr}(F^T L F), \quad (6)$$

where Laplacian matrix  $L = D - Z$ ,  $D$  is a diagonal matrix and its elements are the column sums of  $Z$ . Combining Eq. (6) with Eq. (5), our proposed Similarity Preserving Clustering (SPC) is formulated as

$$\min_{Z, F} \frac{1}{2} \text{Tr}(K + Z^T K Z) - \alpha \text{Tr}(KZ) + \beta \text{Tr}(F^T L F) + \gamma \|Z\|_F^2 \text{ s.t. } F^T F = I, Z \geq 0. \quad (7)$$

By solving Eq. (7), we can obtain a structured graph  $Z$ , which has exactly  $c$  connected components. By running Matlab built-in

---

#### Algorithm 1 Similarity Preserving Clustering (SPC).

---

**Input:** Kernel matrix  $K$ , parameters  $\alpha > 1$ ,  $\beta > 0$ ,  $\gamma > 0$ .

**Initialize:** Random matrix  $Z$ .

**REPEAT**

- 1: Update  $F$  by solving (8).
- 2: For each  $i$ , update the  $i$ th column of  $Z$  according to Eq. (10).
- 3: Project  $Z$  by  $Z = \max(Z, 0)$ .

**UNTIL** stopping criterion is met.

---

function `graphconncomp`, we can obtain which component each sample belongs to.

#### 3.1. Optimization

The problem (7) can be easily solved with an alternating optimization approach. When  $Z$  is fixed, Eq. (7) becomes

$$\min_F \text{Tr}(F^T L F) \text{ s.t. } F^T F = I. \quad (8)$$

It is quite standard to achieve  $F$  which is formed by the  $c$  eigenvectors of  $L$  corresponding to the  $c$  smallest eigenvalues.

When  $F$  is fixed, Eq. (7) can be written column-wisely

$$\min_{Z_i} \frac{1}{2} Z_i^T K Z_i - \alpha K_{i,:} Z_i + \frac{\beta}{2} d_i^T Z_i + \gamma Z_i^T Z_i, \quad (9)$$

where  $d_i = \sum_{j=1}^n \|F_i - F_j\|^2$  and we have used equality  $\sum_{i,j} \frac{1}{2} \|F_i - F_j\|^2 Z_{ij} = \text{Tr}(F^T L F)$ . It is easy to achieve the closed-form solution

$$Z_i = (K + 2\gamma I)^{-1} \left( \alpha K_{i,:} - \frac{\beta d_i}{2} \right) \quad (10)$$

Once parameter  $\gamma$  is given,  $(K + 2\gamma I)$  becomes a constant. Therefore, we only perform the matrix inversion once. We summarize the steps in Algorithm 1. Our algorithm stops if the maximum iteration number 200 is reached or the relative change of  $Z$  is less than  $10^{-5}$ .

### 4. Multiple kernel learning

Different kernels correspond to different notions of similarity and lead to different results. This makes it not be reliable for practical applications. Multiple Kernel Learning (MKL) offers a principled way to encode complementary information and automatically learning an optimal combination of distinct kernels [48,49]. Instead of heuristic kernel selection, a principled method is developed to automatically learn a good combination of multiple kernels.

Specifically, suppose there are in total  $r$  kernels, we introduce kernel weight for each kernel, e.g.,  $w_i$  for kernel  $K^i$ . We denote the combined kernel as  $H = \sum_{i=1}^r w_i K^i$ , and the weight distribution satisfies  $\sum_{i=1}^r \sqrt{w_i} = 1$  [50]. Finally, our multiple kernel learning based similarity preserving clustering (mSPC) method can be formulated as

$$\min_{Z, F, w} \frac{1}{2} \text{Tr}(H + Z^T H Z) - \alpha \text{Tr}(H Z) + \beta \text{Tr}(F^T L F) + \gamma \|Z\|_F^2 \text{ s.t. } F^T F = I, Z \geq 0.$$

$$H = \sum_{i=1}^r w_i K^i, \sum_{i=1}^r \sqrt{w_i} = 1, w_i \geq 0. \quad (11)$$

The problem (11) can be solved in a similar way as (7). In specific, we repeat the following steps.

- (1) Updating  $Z$  and  $F$  when  $w$  is fixed: we can directly obtain  $H$ , and the optimization problem (11) is identical to Eq. (7). We implement Algorithm 1 with  $H$  as the input kernel matrix.
- (2) Updating  $w$  when  $Z$  and  $F$  are fixed: solving Eq. (11) with respect to  $w$  can be reformulated as

$$\min_{\mathbf{w}} \sum_{i=1}^r w_i h_i \quad \text{s.t.} \quad \sum_{i=1}^r \sqrt{w_i} = 1, w_i \geq 0, \quad (12)$$

where

$$h_i = \text{Tr}(K^i - 2\alpha K^i Z + Z^T K^i Z). \quad (13)$$

The Lagrange function of Eq. (12) is

$$\mathcal{J}(\mathbf{w}) = \mathbf{w}^T \mathbf{h} + \eta \left( 1 - \sum_{i=1}^r \sqrt{w_i} \right). \quad (14)$$

By utilizing the Karush–Kuhn–Tucker (KKT) condition with  $\frac{\partial \mathcal{J}(\mathbf{w})}{\partial w_i} = 0$  and the constraint  $\sum_{i=1}^r \sqrt{w_i} = 1$ , we have  $\mathbf{w}$  as follows:

$$w_i = \left( h_i \sum_{j=1}^r \frac{1}{h_j} \right)^{-2}. \quad (15)$$

The complete procedure is summarized in Algorithm 2.

**Algorithm 2** The algorithm of mSPC.

**Input:** A set of kernel matrices  $\{K^i\}_{i=1}^r$ , parameters  $\alpha > 1$ ,  $\beta > 0$ ,  $\gamma > 0$ .

**Initialize:** Random matrix  $Z$ ,  $w_i = 1/r$ .

**REPEAT**

- 1: Calculate  $H$ .
- 2: Update  $F$  by performing singular value decomposition on  $L = D - Z$  and finding the  $c$  smallest eigenvectors.
- 3: Update  $Z$  column-wisely according to (10).
- 4: Update  $\tilde{\mathbf{h}}$  by (13).
- 5: Calculate  $\tilde{\mathbf{w}}$  according to (15).

**UNTIL** stopping criterion is met.

## 5. Experiment

In this section, we perform extensive experiments to demonstrate the effectiveness of our proposed models.

### 5.1. Experiment on synthetic data

We generate a synthetic data set with 300 points. The data points distribute in the pattern of two moons. Each moon is considered as a cluster. In Fig. 1, we present the clustering results of our proposed SPC and standard k-means. Gaussian kernel with  $t = 10$  is used in our SPC model. We can observe that our method performs much better than k-means. We quantitatively assess the clustering performance in terms of accuracy (Acc), normalized mutual information (NMI), and Purity. For SPC, Acc, NMI, and Purity are 93%, 63.49%, 93%, respectively. Correspondingly, k-means produces 73.67%, 16.87%, and 73.67%.

### 5.2. Experiment on real data

#### 5.2.1. Data sets

We conduct our experiments with eight benchmark data sets, which are widely used in clustering experiments. We show the statistics of these data sets in Table 1.

These data sets are from different fields. Specifically, YALE<sup>1</sup>, JAFFE<sup>2</sup>, ORL<sup>3</sup> are three face databases. Each image represents different facial expressions or configurations due to times, illumination conditions, and glasses/no glasses. Fig. 2(a) and (b) shows

**Table 1**  
Description of the data sets.

	# instances	# features	# classes
YALE	165	1024	15
JAFFE	213	676	10
ORL	400	1024	40
YEAST	1484	1470	10
USPS	1854	256	20
TR11	414	6429	9
TR41	878	7454	10
TR45	690	8261	10

some example images from JAFFE and YALE database, respectively. YEAST is a microarray data set. USPS data set<sup>4</sup> is obtained from the scanning of handwritten digits from envelopes by the U.S. Postal Service. Some sample digits are shown in Fig. 2(c). The last three data sets in Table 1 are text data<sup>5</sup>.

We manually construct 12 kernels. They consist of seven Gaussian kernels  $K(x, y) = \exp(-\|x - y\|_2^2 / (td_{\max}^2))$  with  $t \in \{0.01, 0.05, 0.1, 1, 10, 50, 100\}$  and  $d_{\max}$  denotes the maximal distance between data points; four polynomial kernels  $K(x, y) = (a + x^T y)^b$  of the form with  $a \in \{0, 1\}$  and  $b \in \{2, 4\}$ ; a linear kernel  $K(x, y) = x^T y$ . Furthermore, all kernel matrices are normalized to  $[0, 1]$  range to avoid numerical inconsistency.

#### 5.2.2. Comparison methods

To fully investigate the performance of our method on clustering, we choose a good set of methods to compare. In general, they can be classified into two categories: graph-based and kernel-based clustering methods.

- *Spectral Clustering (SC)* [15]. We use kernel matrix as its graph input. For our SPC, we learn graph from kernels.
- *Robust Kernel K-means (RKKM)*<sup>6</sup>[14]. As an extension to the classical K-means clustering method, RKKM has the capability of dealing with nonlinear structures, noise, and outliers in the data. We also compare with its multiple kernel learning version: RMKKM.
- *Simplex Sparse Representation (SSR)* [51]. Based on self-expression, SSR achieves satisfying performance in numerous data sets.
- *Clustering with Adaptive Neighbor (CAN)* [22]. Based on adaptive local structure learning, CAN constructs the similarity graph by Eq. (1).
- *Kernel Spectral Clustering (KSC)* [20]. Based on a weighted kernel principal component analysis strategy, KSC performs multiway spectral clustering. Moreover, Balanced Line Fit (BLF) is proposed to obtain model parameters.
- Our proposed SPC and mSPC<sup>7</sup>. Our proposed single kernel and multiple kernel learning based similarity preserving clustering methods.
- *SPC1* and *mSPC1*. To observe the effect of similarity preserving, we let  $\alpha = 1$  in SPC and name this method as SPC1. Similarly, we have mSPC1. They are equivalent to the methods in [34].
- *Multiple Kernel K-means (MKKM)*<sup>8</sup>[52]. The MKKM extends K-means in a multiple kernel setting. It imposes a different constraint on the kernel weight distribution.

<sup>4</sup> <http://www-stat.stanford.edu/~tibs/ElemStatLearn/data.html>.

<sup>5</sup> <http://www-users.cs.umn.edu/~han/data/tmdata.tar.gz>.

<sup>6</sup> <https://github.com/csliangdu/RMKKM>.

<sup>7</sup> <https://github.com/sckangz/SPC>.

<sup>8</sup> <http://imp.iis.sinica.edu.tw/IVCLab/research/Sean/mkfc/code>.

<sup>1</sup> <http://vision.ucsd.edu/content/yale-face-database>.

<sup>2</sup> <http://www.kasrl.org/jaffe.html>.

<sup>3</sup> <http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html>.

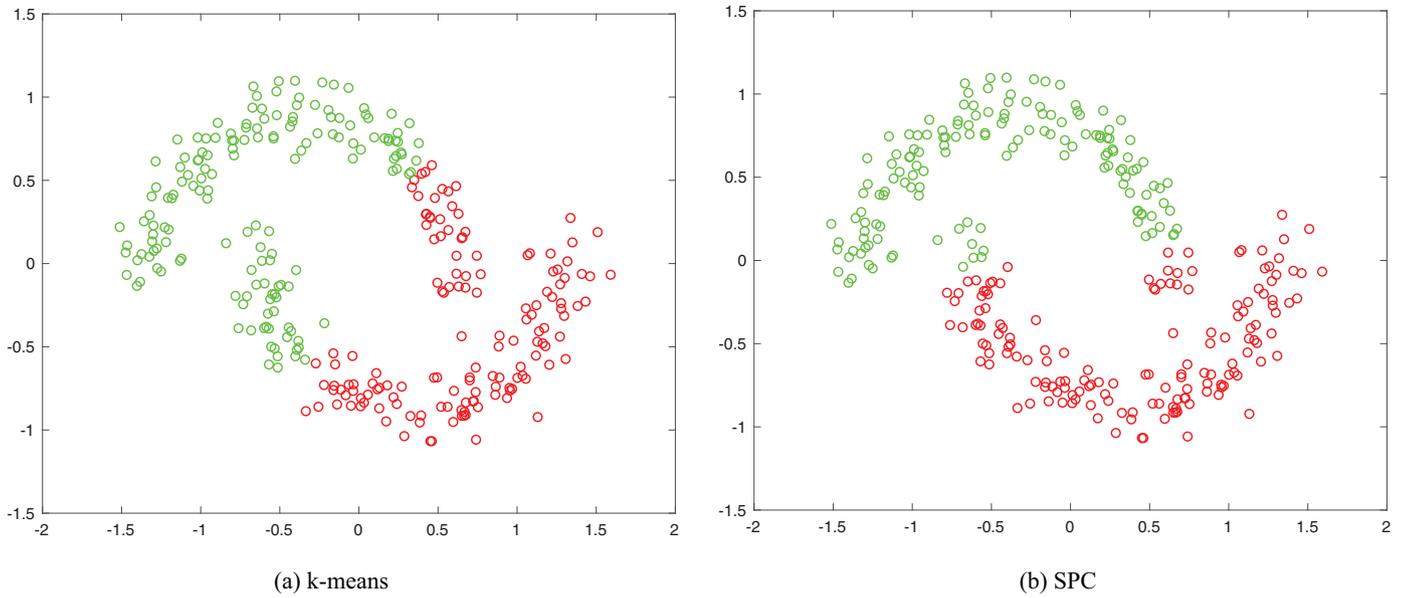


Fig. 1. The clustering results on synthetic data.

Table 2

Clustering results measured on benchmark data sets. The average performance on those 12 kernels are put in parenthesis. For KSC, we run 10 times and report the best performance and their mean value. The best results for single and multiple kernel methods are highlighted in boldface.

Data	SC	RKKM	SSR	CAN	SPC1	SPC	KSC	MKKM	AASC	RMKKM	mSPC1	mSPC
(a) Accuracy(%)												
YALE	49.42(40.52)	48.09(39.71)	54.55	58.79	55.29(45.07)	<b>60.60</b> (46.95)	36.36(29.76)	45.70	40.64	52.18	56.97	<b>63.03</b>
JAFFE	74.88(54.03)	75.61(67.98)	87.32	98.12	97.18(86.55)	<b>98.03</b> (86.20)	72.77(66.48)	74.55	30.35	87.07	97.18	<b>98.12</b>
ORL	57.96(46.65)	54.96(46.88)	69.00	61.50	62.47(50.64)	<b>75.75</b> (52.48)	37.00(32.70)	47.51	27.20	55.60	65.25	<b>75.43</b>
YEAST	35.55(30.89)	34.04(31.52)	29.99	34.62	35.72(30.55)	<b>37.85</b> (31.65)	31.19(28.36)	13.04	35.38	31.63	36.08	<b>39.15</b>
USPS	35.18(26.90)	65.16(55.72)	64.83	62.84	65.94(56.38)	<b>67.25</b> (56.94)	61.49(47.87)	63.72	37.36	65.47	65.32	<b>68.74</b>
TR11	50.98(43.32)	53.03(45.04)	41.06	38.89	70.88(54.25)	<b>78.26</b> (56.37)	50.73(41.86)	50.13	47.15	57.71	73.43	<b>79.63</b>
TR41	63.52(44.80)	56.76(46.80)	63.78	62.87	67.28(52.75)	<b>72.89</b> (49.40)	53.42(45.11)	56.10	45.90	62.65	67.31	<b>80.41</b>
TR45	57.39(45.96)	58.13(45.69)	71.45	56.96	73.59(53.06)	<b>75.07</b> (57.26)	53.48(44.99)	58.46	52.64	64.00	74.35	<b>75.64</b>
(b) NMI(%)												
YALE	52.92(44.79)	52.29(42.87)	57.26	57.67	56.37(45.00)	<b>61.32</b> (45.62)	44.84(35.47)	50.06	46.83	55.58	56.52	<b>61.36</b>
JAFFE	82.08(59.35)	83.47(74.01)	92.93	97.31	96.35(84.67)	<b>98.62</b> (83.30)	73.67(68.39)	79.79	27.22	89.37	95.61	<b>97.36</b>
ORL	75.16(66.74)	74.23(63.91)	84.23	76.59	79.36(63.98)	<b>86.06</b> (66.56)	56.78(54.21)	68.86	43.77	74.83	80.04	<b>85.93</b>
YEAST	<b>21.38</b> (6.18)	17.27(9.31)	15.85	20.06	15.37(9.62)	16.44(10.25)	13.87(13.83)	10.29	<b>21.19</b>	20.71	15.89	16.23
USPS	29.71(21.33)	63.94(52.90)	72.68	76.13	74.85(55.28)	<b>78.54</b> (58.93)	48.45(40.27)	62.25	29.81	63.60	75.29	<b>79.88</b>
TR11	43.11(31.39)	49.69(33.48)	27.60	19.17	58.14(37.42)	<b>63.10</b> (35.94)	46.22(36.51)	44.56	39.39	56.08	60.15	<b>63.90</b>
TR41	61.33(36.60)	60.77(40.86)	59.56	51.13	65.90(43.28)	<b>71.22</b> (38.54)	44.21(39.19)	57.75	43.05	63.47	65.11	<b>70.50</b>
TR45	48.03(33.22)	57.86(38.96)	67.82	49.31	74.21(44.29)	<b>75.94</b> (46.28)	43.41(42.95)	56.17	41.94	62.73	<b>74.97</b>	74.57
(c) Purity(%)												
YALE	51.61(43.06)	49.79(41.74)	58.18	59.39	56.79(55.25)	<b>60.53</b> (56.28)	44.85(34.67)	47.52	42.33	53.64	60.00	<b>66.67</b>
JAFFE	76.83(56.56)	79.58(71.82)	96.24	98.12	97.85(96.03)	<b>98.25</b> (97.02)	77.00(72.30)	76.83	33.08	88.90	97.18	<b>98.12</b>
ORL	61.45(51.20)	59.60(51.46)	76.50	68.5	73.28(70.02)	<b>82.08</b> (76.56)	41.00(36.73)	52.85	31.56	60.23	77.00	<b>82.69</b>
YEAST	53.05(35.37)	45.39(38.18)	44.29	58.97	57.38(40.83)	<b>65.72</b> (44.75)	32.21(31.14)	32.58	52.71	33.21	60.27	<b>66.51</b>
USPS	37.48(33.27)	72.49(62.49)	75.84	71.89	76.90(63.78)	<b>77.54</b> (64.82)	62.84(51.48)	70.77	42.40	73.45	77.03	<b>79.25</b>
TR11	58.79(50.23)	67.93(56.40)	85.02	44.20	81.79(80.12)	<b>90.10</b> (83.86)	52.90(46.76)	65.48	54.67	72.93	87.44	<b>93.04</b>
TR41	73.68(56.45)	74.99(60.21)	75.40	67.54	73.05(71.13)	<b>80.67</b> (74.79)	53.42(47.92)	72.83	62.05	77.57	73.69	<b>82.45</b>
TR45	61.25(50.02)	68.18(53.75)	83.62	60.87	78.74(77.82)	<b>86.32</b> (80.03)	55.51(49.29)	69.14	57.49	75.20	78.26	<b>87.59</b>

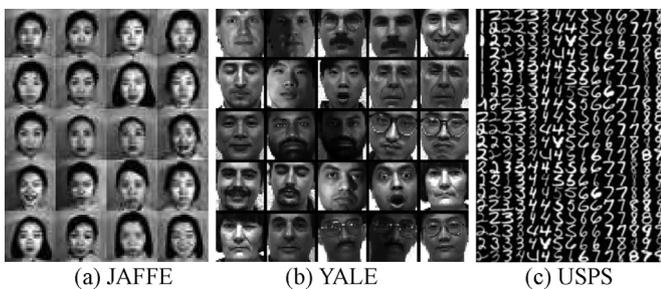


Fig. 2. Sample images of USPS, JAFFE, and YALE.

• *Affinity Aggregation for Spectral Clustering (AASC)*<sup>9</sup>[53]. The AASC is an extension of spectral clustering to the situation when multiple affinities exist.

For a fair comparison, we either use the recommended parameter settings in their respective papers or tune each method to obtain the best performance. In fact, the optimal performance for SC, RKKM, MKKM, AASC, and RMKKM methods can be easily obtained through implementing the package given in [14]. SC, SSR, and CAN are parameter-free models. KSC selects parameters based on Balanced Line Fit principle.

<sup>9</sup> <http://imp.iis.sinica.edu.tw/IVCLab/research/Sean/aasc/code> .

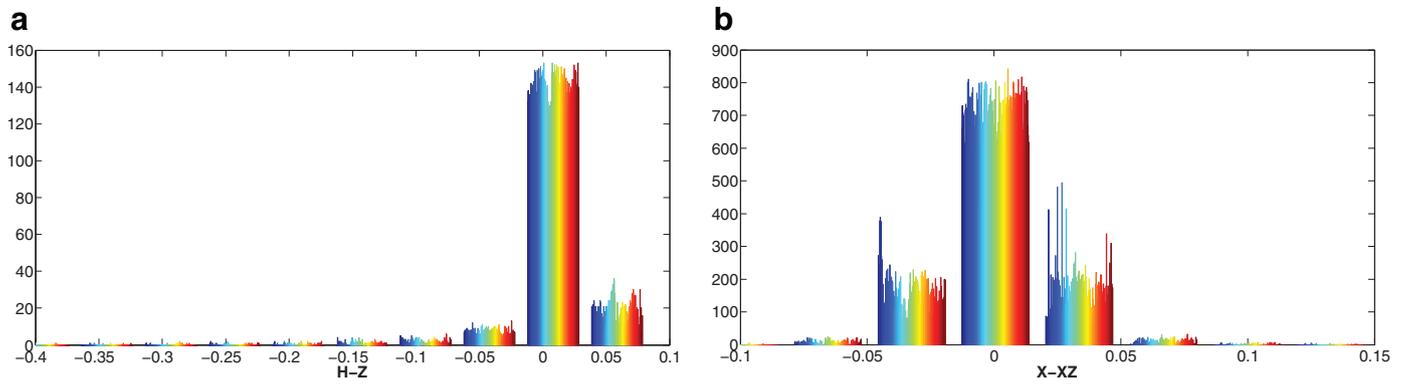


Fig. 3. The visualization of similarity preserving effect.

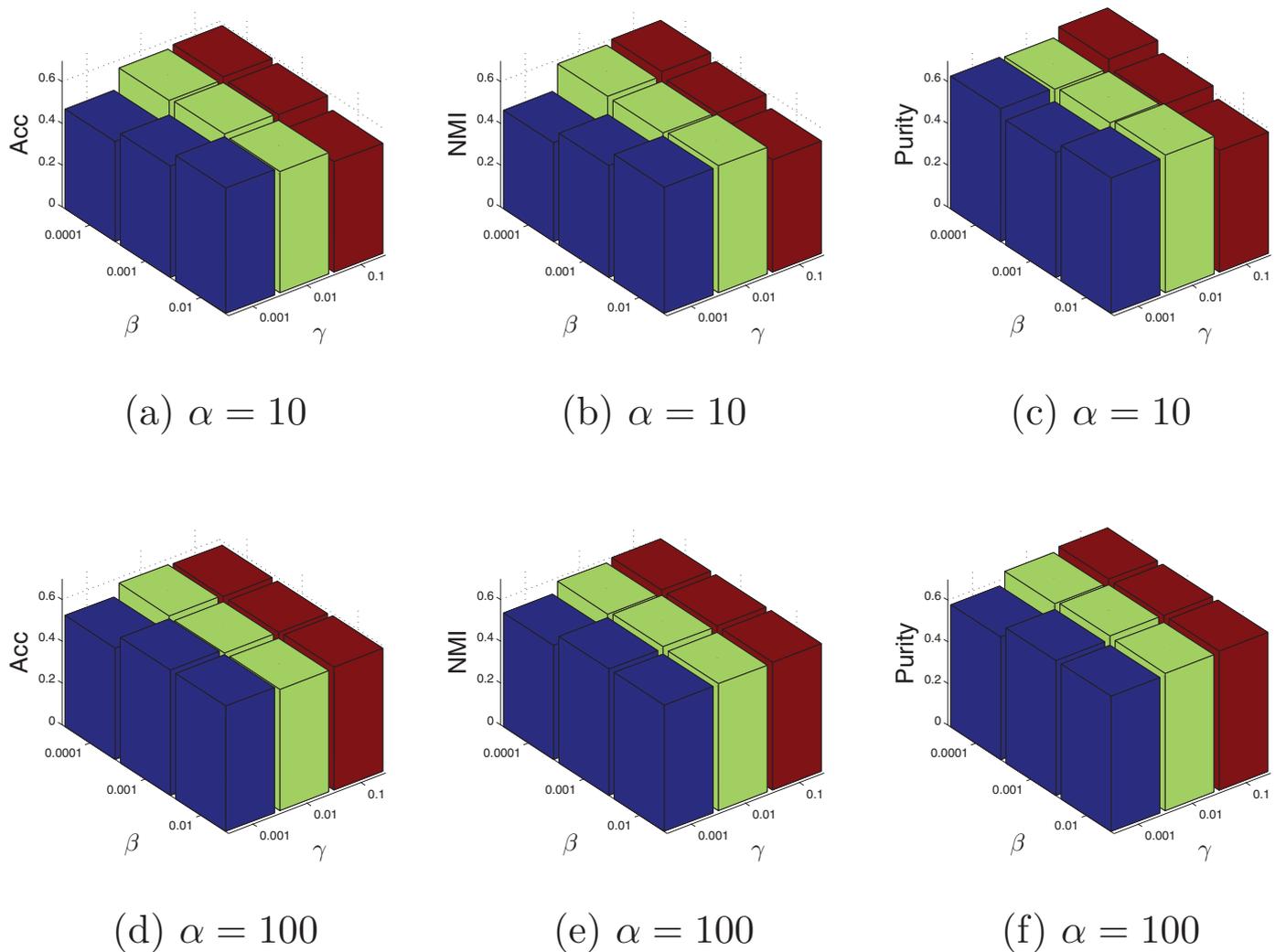


Fig. 4. The influence of parameters on YALE data set.

### 5.2.3. Results

All results are summarized in Table 2. We can see that our methods SPC and mSPC outperform others in most cases. In particular, we have the following observations: (1) the improvement of SPC over SPC1 is considerable. Noted that the only difference between SPC and SPC1 is that SPC explicitly considers the similarity preserving effect. In other words, SPC adds the proposed term Eq. (4), which aims to keep the learned graph matrix  $Z$  close to the kernel matrix  $K$ , so that the similarity information carried

by the kernel matrix will transfer to the learned graph matrix. Hence this demonstrates the significance of similarity preserving in graph learning; (2) for multiple kernel methods, mSPC also performs better than mSPC1 in most experiments. This once again confirms the importance of similarity preserving; (3) compared to self-expression based method SSR, our advantage is also obvious. For example, in TR11, SPC enhances the accuracy from 41.06% to 78.26%. Note that our basic objective function Eq. (3) is also derived from self-expression idea. However, our method is kernel

method; (4) with respect to traditional spectral clustering, kernel spectral clustering, the recently proposed robust kernel K-means method, adaptive local structure graph learning method, the improvement is very promising; (5) in terms of multiple kernel learning approach, mSPC also achieves much better performance than other state-of-the-art techniques.

To better illustrate the effect of similarity preserving, we visualize the results of YALE data in Fig. 3. In specific, Fig. 3 plots the histogram of  $H - Z$ , i.e., the difference between the learned kernel in Eq. (11) and similarity matrix. We can see that they are quite close for most elements and the difference is the refinement brought by our learning algorithm. The manually constructed kernel matrix often fails to reflect the underlying relationships among samples due to the inherent noise or the inappropriate use of a metric function. This is validated by the experimental results. Note that for SC method, we directly treat kernel matrix as similarity matrix, while for our proposed SPC method, we use the learned similarity matrix  $Z$  to perform clustering. It can be seen that the results of SPC are much better than that of SC.

Fig. 3 displays the difference between the original data  $X$  and the reconstructed data  $XZ$ . Good reconstruction means that  $Z$  represents the similarity pretty well. The reconstruction error accounts for noise or outliers in the original data. As shown by Fig. 3, our learned  $Z$  reconstructs the original data with a small error. Therefore, our proposed approach can achieve a high-quality similarity matrix.

#### 5.2.4. Parameter analysis

As shown in Eq. (11), there are three parameters in our model. As we mentioned previously,  $\alpha$  is bigger than one. Take YALE data set as an example, we demonstrate the sensitivity of our model mSPC in Fig. 4. We can see that it works well over a wide range of values. Note that  $\alpha = 1$  case has been discussed by SPC1 and mSPC1 methods in Table 2. When  $\alpha = 1$ , Eqs. (7) and (11) do not possess similarity preserving capability.

## 6. Conclusion

In this paper, we propose a clustering algorithm which can exploit similarity information of raw data. Furthermore, the structure information of a graph is also considered in our objective function. Comprehensive experimental results on real data sets well demonstrate the superiority of the proposed method on the clustering task. It has been shown that the performance of the proposed method is largely determined by the choice of the kernel function. To this end, we develop a multiple kernel learning method, which is capable of automatically learning an appropriate kernel from a pool of candidate kernels. In the future, we will examine the effectiveness of our framework on the semi-supervised learning task.

#### Declarations of interest

None.

#### Acknowledgment

This paper was in part supported by Grants from the Natural Science Foundation of China (Nos. 61806045, 61572111, and 61872062), three Fundamental Research Fund for the Central Universities of China (Nos. ZYGX2017KYQD177, A03017023701012, and ZYGX2016J086), and a 985 Project of UESTC (No. A1098531023601041).

## References

- [1] A.K. Jain, M.N. Murty, P.J. Flynn, Data clustering: a review, *ACM Comput. Surv. (CSUR)* 31 (3) (1999) 264–323.
- [2] Y. Yang, F.M. Shen, Z. Huang, H.T. Shen, X.L. Li, Discrete nonnegative spectral clustering, *IEEE Trans. Knowl. Data Eng.* 29 (9) (2017) 1834–1845.
- [3] S. Huang, Z. Kang, Z. Xu, Self-weighted multi-view clustering with soft capped norm, *Knowl.-Based Syst.* 158 (2018) 1–8.
- [4] X. Chen, Y. Ye, X. Xu, J.Z. Huang, A feature group weighting method for subspace clustering of high-dimensional data, *Pattern Recognit.* 45 (1) (2012) 434–446.
- [5] Y. Ren, K. Hu, X. Dai, L. Pan, S.C. Hoi, Z. Xu, Semi-supervised deep embedded clustering, *Neurocomputing* 325 (2019) 121–130.
- [6] R. Xu, D. Wunsch, Survey of clustering algorithms, *IEEE Trans. Neural Netw.* 16 (3) (2005) 645–678.
- [7] S. Huang, Z. Kang, I.W. Tsang, Z. Xu, Auto-weighted multi-view clustering via kernelized graph learning, *Pattern Recognit.* 88 (2019) 174–184.
- [8] X. Yang, W. Yu, R. Wang, G. Zhang, F. Nie, Fast spectral clustering learning with hierarchical bipartite graph for large-scale data, *Pattern Recognit. Lett.* (2018).
- [9] 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. (TIST)* 9 (5) (2018) 57.
- [10] S. Lloyd, Least squares quantization in PCM, *IEEE Trans. Inf. Theory* 28 (2) (1982) 129–137.
- [11] B. Yang, X. Fu, N.D. Sidiropoulos, M. Hong, Towards k-means-friendly spaces: simultaneous deep learning and clustering, in: *Proceedings of the International Conference on Machine Learning*, 2017, pp. 3861–3870.
- [12] X. Chen, X. Xu, Y. Ye, J.Z. Huang, TW-k-means: automated two-level variable weighting clustering algorithm for multi-view data, *IEEE Trans. Knowl. Data Eng.* 25 (4) (2013) 932–944.
- [13] B. Schölkopf, A. Smola, K.-R. Müller, Nonlinear component analysis as a kernel eigenvalue problem, *Neural Comput.* 10 (5) (1998) 1299–1319.
- [14] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, Y.-D. Shen, Robust multiple kernel k-means using  $\ell_{21}$ -norm, in: *Proceedings of the Twenty-fourth International Conference on Artificial Intelligence*, AAAI Press, 2015, pp. 3476–3482.
- [15] A.Y. Ng, M.I. Jordan, Y. Weiss, et al., On spectral clustering: analysis and an algorithm, *Adv. Neural Inf. Process. Syst.* 2 (2002) 849–856.
- [16] Z. Kang, C. Peng, Q. Cheng, Z. Xu, Unified spectral clustering with optimal graph, in: *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence*, 2018.
- [17] D.H. Peluffo-Ordóñez, M.A. Becerra, A.E. Castro-Ospina, X. Blanco-Valencia, J.C. Alvarado-Pérez, R. Therón, A. Anaya-Isaza, On the relationship between dimensionality reduction and spectral clustering from a kernel viewpoint, in: *Proceedings of the Thirteenth International Conference Distributed Computing and Artificial Intelligence*, Springer, 2016, pp. 255–264.
- [18] U. Von Luxburg, A tutorial on spectral clustering, *Stat. Comput.* 17 (4) (2007) 395–416.
- [19] L. Zelnik-Manor, P. Perona, Self-tuning spectral clustering, *Adv. Neural Inf. Process. Syst.* 17 (2005) 1601–1608.
- [20] C. Alzate, J.A. Suykens, Multiway spectral clustering with out-of-sample extensions through weighted kernel PCA, *IEEE Trans. Pattern Anal. Mach. Intell.* 32 (2) (2010) 335–347.
- [21] R. Langone, R. Mall, C. Alzate, J.A. Suykens, Kernel spectral clustering and applications, in: *Unsupervised Learning Algorithms*, Springer, 2016, pp. 135–161.
- [22] 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.
- [23] B. Cheng, J. Yang, S. Yan, Y. Fu, T.S. Huang, Learning with l1-graph for image analysis, *Trans. Image Process.* 19 (4) (2010) 858–866, doi:10.1109/TIP.2009.2038764.
- [24] E. Elhamifar, R. Vidal, Sparse subspace clustering: algorithm, theory, and applications, *IEEE Trans. Pattern Anal. Mach. Intell.* 35 (11) (2013) 2765–2781.
- [25] 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.
- [26] C. Tang, X. Zhu, X. Liu, M. Li, P. Wang, C. Zhang, L. Wang, Learning joint affinity graph for multi-view subspace clustering, *IEEE Trans. Multimed.* (2018).
- [27] Z. Kang, C. Peng, Q. Cheng, Kernel-driven similarity learning, *Neurocomputing* 267 (2017) 210–219.
- [28] Z. Kang, L. Wen, W. Chen, Z. Xu, Low-rank kernel learning for graph-based clustering, *Knowl.-Based Syst.* 163 (2019) 510–517.
- [29] B.D. Haeffele, R. Vidal, Structured low-rank matrix factorization: global optimality, algorithms, and applications, arXiv:1708.07850 (2017).
- [30] Z. Kang, Y. Lu, Y. Su, C. Li, Z. Xu, Similarity learning via kernel preserving embedding, in: *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, AAAI Press, 2019.
- [31] Z. Zhao, L. Wang, H. Liu, J. Ye, On similarity preserving feature selection, *IEEE Trans. Knowl. Data Eng.* 25 (3) (2013) 619–632.
- [32] H. Cai, V.W. Zheng, K.C.-C. Chang, A comprehensive survey of graph embedding: problems, techniques, and applications, *IEEE Trans. Knowl. Data Eng.* 30 (9) (2018) 1616–1637.
- [33] Z. Kang, H. Pan, S.C. Hoi, Z. Xu, Robust graph learning from noisy data, *IEEE Trans. Cybern.* (2019).

- [34] 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 (AAAI-17), AAAI Press, 2017.
- [35] X. Niyogi, Locality preserving projections, in: Neural Information Processing Systems, 16, MIT, 2004, p. 153.
- [36] L. Zhang, Q. Zhang, B. Du, J. You, D. Tao, Adaptive manifold regularized matrix factorization for data clustering, in: Proceedings of the Twenty-sixth international joint conference on artificial intelligence, 2017, pp. 3405–33999.
- [37] S. Huang, Z. Xu, J. Lv, Adaptive local structure learning for document co-clustering, *Knowl.-Based Syst.* 148 (2018) 74–84.
- [38] L. Du, Y.-D. Shen, Unsupervised feature selection with adaptive structure learning, in: Proceedings of the Twenty-first ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, 2015, pp. 209–218.
- [39] F. Nie, G. Cai, X. Li, Multi-view clustering and semi-supervised classification with adaptive neighbours, in: Proceedings of the AAAI, 2017, pp. 2408–2414.
- [40] Y. Yang, Z. Wang, J. Yang, J. Wang, S. Chang, T.S. Huang, Data clustering by Laplacian regularized 11-graph, in: Proceedings of the AAAI, 2014, pp. 3148–3149.
- [41] L. Zhang, M. Yang, X. Feng, Sparse representation or collaborative representation: Which helps face recognition? in: Proceedings of the 2011 IEEE International Conference on Computer vision (ICCV), IEEE, 2011, pp. 471–478.
- [42] L. Zhuang, Z. Zhou, S. Gao, J. Yin, Z. Lin, Y. Ma, Label information guided graph construction for semi-supervised learning, *IEEE Trans. Image Process.* (2017).
- [43] S.T. Roweis, L.K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science* 290 (5500) (2000) 2323–2326.
- [44] X. He, P. Niyogi, Locality preserving projections, 2004, pp. 153–160.
- [45] M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, 2002, pp. 585–591.
- [46] B. Mohar, Y. Alavi, G. Chartrand, O. Oellermann, The Laplacian spectrum of graphs, *Graph Theory Combinator. Appl.* 2 (871–898) (1991) 12.
- [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. Sonnenburg, G. Rätsch, C. Schäfer, B. Schölkopf, Large scale multiple kernel learning, *J. Mach. Learn. Res.* 7 (Jul) (2006) 1531–1565.
- [49] Z. Kang, X. Lu, J. Yi, Z. Xu, Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification, in: Proceedings of the Twenty-seventh International Joint Conference on Artificial Intelligence, AAAI Press, 2018, pp. 2312–2318.
- [50] M. Gönen, E. Alpaydın, Multiple kernel learning algorithms, *J. Mach. Learn. Res.* 12 (Jul) (2011) 2211–2268.
- [51] J. Huang, F. Nie, H. Huang, A new simplex sparse learning model to measure data similarity for clustering, in: Proceedings of the IJCAI, 2015, pp. 3569–3575.
- [52] H.-C. Huang, Y.-Y. Chuang, C.-S. Chen, Multiple kernel fuzzy clustering, *IEEE Trans. Fuzzy Syst.* 20 (1) (2012) 120–134.
- [53] H.-C. Huang, Y.-Y. Chuang, C.-S. Chen, Affinity aggregation for spectral clustering, in: Proceedings of the 2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), IEEE, 2012, pp. 773–780.



**Zhao Kang** received his Ph.D. degree in Computer Science from Southern Illinois University Carbondale, USA, in 2017. Currently, he is an assistant professor at the School of Computer Science and Engineering at University of Electronic Science and Technology of China (UESTC). His research interests are machine learning, data mining, pattern recognition, and deep learning. He has published over 30 research papers in top-tier conferences and journals, including AAAI, IJCAI, ICDE, CVPR, SIGKDD, ICDM, CIKM, SDM, IEEE Transactions on Cybernetics, ACM TIST, ACM TKDD, Pattern Recognition, Neurocomputing, Knowledge-Based Systems. He has been a PC member or reviewer to a number of top conferences such as AAAI, IJCAI, ICCV, MM, ICDM, etc. He regularly servers as a reviewer to IEEE TNNLS, IEEE Transactions on Cybernetics, Neurocomputing, etc.

**Honghui Xu** is a undergraduate student in the School of Computer Science and Engineering, University of Electronic Science and Technology of China.



**Boyu Wang** is an assistant professor at the Department of Computer Science at the University of Western Ontario. He was a postdoctoral fellow at the University of Pennsylvania and Princeton University. Before that, he obtained his PhD in Computer Science at McGill University in 2016. His research interests include machine learning, brain signal analysis, neuroscience, and biomedical engineering.



**Hongyuan Zhu** is currently a Research Scientist with the Institute for Infocomm Research, A\*STAR, Singapore. His research interests include multimedia content analysis and segmentation, specially image segmentation/co-segmentation, object detection, scene recognition, and saliency detection.



**Zenglin Xu** is a Professor in School of Computer Science and Engineering at University of Electronic Science and Technology of China (UESTC). He is the founder and director of the Statistical Machine Intelligence and Learning (SMILE) Lab. He obtained his Ph.D. in Computer Science and Engineering from the Chinese University of Hong Kong. His research interest includes machine learning and its applications on social network analysis, health informatics, and cyber security analytics. He has published over 50 papers in prestigious journals and conferences such as NIPS, ICML, IJCAI, AAAI, IEEE PAMI, IEEE TNN, etc. He is also the recipient of the APNNS young researcher award, and the best student paper honorable mention of AAAI 2015. Dr. Xu has been a PC member or reviewer to a number of top conferences such as NIPS, ICML, AAAI, IJCAI, etc. He regularly servers as a reviewer to IEEE TPAMI, JMLR, PR, IEEE TNNLS, IEEE TKDE, ACM TKDD, etc.