Robust graph learning from noisy data

Zhao KANG

Haiqi PAN

Steven C. H. HOI
Singapore Management University, chhoi@smu.edu.sg

Zenglin XU

Robust Graph Learning from Noisy Data

Zhao Kang*, Haiqi Pan, Steven C.H. Hoi†, Zenglin Xu

December 18, 2018

Abstract

Learning graphs from data automatically has shown encouraging performance on clustering and semisupervised learning tasks. However, real data are often corrupted, which may cause the learned graph to be inexact or unreliable. In this paper, we propose a novel robust graph learning scheme to learn reliable graphs from real-world noisy data by adaptively removing noise and errors in the raw data. We show that our proposed model can also be viewed as a robust version of manifold regularized robust PCA, where the quality of the graph plays a critical role. The proposed model is able to boost the performance of data clustering, semisupervised classification, and data recovery significantly, primarily due to two key factors: 1) enhanced low-rank recovery by exploiting the graph smoothness assumption, 2) improved graph construction by exploiting clean data recovered by robust PCA. Thus, it boosts the clustering, semi-supervised classification, and data recovery performance overall. Extensive experiments on image/document clustering, object recognition, image shadow removal, and video background subtraction reveal that our model outperforms the previous state-of-the-art methods.

1 Introduction

Graph structure is widely used to represent data in real-world. Consequently, graph construction becomes a fundamental problem in machine learning, pattern recognition, and data mining [1–5]. Moreover, it is crucial to the performance of many algorithms. In this paper, we focus on two basic applications: clustering [6–11] and semisupervised classification [12–17]. Due to their promising performance, a variety of graph-based clustering and classification methods have been developed during the past decades.

Clustering partitions the data objects into different groups based on certain property [18–20]. As a classical task, a number of clustering techniques have

---

*Z. Kang, H. Pan, and Z. Xu are with the School of Computer Science and Engineering, University of Electronic Science and Technology of China, Chengdu, Sichuan 611731, China. E-mail: {zkang, zlxu}@uestc.edu.cn.

†S. C.H. Hoi is with School of Information Systems, Singapore Management University,Singapore, Singapore Singapore 17890. E-mail: chhoi@smu.edu.sg

been developed. Among them, two most successful techniques are k-means and spectral clustering. Compared with k-means and its extensions [21–27], spectral clustering method presents more capability in detecting complex structures of data [28–32]. It works by embedding the data point into a vector space that is spanned by the spectrum of the graph matrix. Therefore, the quality of the graph is crucial to the performance of spectral clustering algorithm. Typically, the graph is constructed by some computation criterion from raw data [33], where the Gaussian kernel function or k-nearest-neighbor approaches are often adopted. However, how to choose an appropriate scaling factor σ or a proper neighbor number k are still open questions. More importantly, both of them highly influence the clustering results [34].

On the other hand, semisupervised learning deals with the situation that a small portion of samples has been labeled [35]. In specific, semisupervised classification aims to predict labels for a large number of unlabeled data points based on the relationships among the data points. A lot of work on this task is based on graph regularizer. In general, the first step is to create a graph with the labeled and unlabeled samples as the vertices and with the edge weights encoding the similarity between the samples. Then, the missing labels are inferred based on the local smoothness over the graph, i.e., the more similar two objects are, the more likely they have similar labels. Consequently, this approach's performance depends greatly upon the edge weights of the graph. Although label inference has been thoroughly investigated, graph construction has attracted much less attention until recent years [14]. There is still no typically used strategy to address this graph construction challenge.

Recently, learning graphs from data automatically has drawn significant attention [36–43]. By this means, the most informative neighbors for each data point are automatically selected and it is free of similarity measure metrics, which are often data-dependent and sensitive to noise and outliers [44]. As a result, the resulted accuracy is much higher than that of traditional manually constructed graphs. Generally speaking, it can be classified into two categories. The first approach is based on self-expressiveness. Basically, it represents every data point by a linear combination of all other data points and the achieved coefficient matrix is seen as the graph matrix. It has demonstrated impressive performance in a number of applications, including clustering [45–48], semisupervised learning [49, 50].

The other one is adaptive neighbors approach. It builds graph by assigning a probability for each data point as the neighborhood of another data point and the learned probability is treated as the similarity between two data points [51]. This approach is supposed to capture the local structure of data. Due to its simplicity and effectiveness, it has been widely utilized in various domains, such as clustering [51], nonnegative matrix factorization [52], feature selection [53,54], multiview learning [55]. However, all these work ignore the fact that the graph is learned from the raw data, which is often contaminated in reality. Consequently, the achieved graph might be inexact or sub-optimal and fails to characterize the true relationships between data points.

In this work, we propose a novel robust graph learning scheme to learn robust

and reliable graphs from real data. Specifically, we decompose the original data into a low-rank matrix D ("clean data") and a sparse matrix E ("noise/errors"). Then we can build the graph S on the clean data D by using adaptive neighbors approach. Instead of following a two steps approach, we propose a joint learning scheme to optimize S and D simultaneously, such that they are mutually enhanced via an alternating optimization approach. Therefore, besides constructing graph for clustering, our model simultaneously recovers the low-rank clean data, i.e., functioning as robust principle component analysis (RPCA) [56]. In summary, the main contributions of our work are:

- We propose a robust graph learning scheme by decomposing the raw data into a clean part and an error part. Rather than adopting a two steps approach, we develop a unified model and jointly learn the graph as well as the clean data. To the best of our knowledge, this is the first work to investigate robust adaptive neighbor graph construction.

- We show that our proposed method can significantly boost a graph Laplacian regularized RPCA model. Unlike many other heuristic methods for constructing graph Laplacian directly from noisy data, our graph Laplacian is adaptively built from clean data. Benefiting from the incorporation of graph smoothness assumption, the quality of the recovered low-rank matrix is enhanced.

- Extensive experiments on face clustering, document clustering, face/object recognition, face image shadow removal, and background separation demonstrate the effectiveness of the proposed technique.

In the remainder of this paper, we introduce the graph learning in Section 2. Then a robust graph learning technique is proposed in Section 3, where the details of the algorithm are also provided. Section 4 presents experimental evaluation on clustering task. The semisupervised application is discussed in Section 5. We compare the data recovery effects in Section 6. Finally, conclusions are drawn in Section 7.

2 Adaptive Neighbor Graph Learning Revisited

We first introduce the notations that are used throughout the rest of the paper. 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 ℓ_1-norm of X is denoted by $\|X\|_1 = \sum_{ij} |x_{ij}|$. The definition of X's nuclear norm is $\|X\|_* = \sum_i \sigma_i$, where σ_i is the i-th singular value of X. $\mathbf{1}$ is a vector with all entries as 1. We use $0 \leq S \leq 1$ to represent all elements in S are in the range of $[0, 1]$. $(s)_+ = max(s, 0)$.

As locality preserving projection (LPP) [57] does, we can assign a probability s_{ij} for x_j as the neighborhood of x_i. Thus, s_{ij} characterizes the similarity between x_i and x_j in some sense. Smaller distance $\|x_i - x_j\|^2$ indicates that x_i

and x_j are quite similar, thus bigger value s_{ij} is. To achieve graph S, we can solve the following problem:

$$\min_{s_i} \sum_{j=1}^n (\frac{1}{2}\|x_i - x_j\|^2 s_{ij} + \gamma s_{ij}^2),$$

$$s.t. \ s_i^T \mathbf{1} = 1, \ 0 \le s_{ij} \le 1, \tag{1}$$

where γ is a trade-off parameter. By defining graph Laplacian matrix $L = D - \frac{S+S^T}{2}$, where D is a diagonal matrix with $d_{ii} = \sum_j \frac{s_{ij}+s_{ji}}{2}$, problem (1) becomes:

$$\min_S Tr(XLX^T) + \gamma \|S\|_F^2,$$

$$s.t. \ S\mathbf{1} = \mathbf{1}, \ 0 \le S \le 1. \tag{2}$$

By optimizing above problem, one can learn S adaptively from the data. Compared with LPP, where neighborhood relationship is predetermined, the structure is automatically learned from the data in Eq. (2). This strategy has attracted great attention recently. [51] performed clustering based on the learned graph. [52] employed the obtained graph to construct the Laplacian graph used in graph-based Nonnegative Matrix Factorization (NMF). [53, 58] presented a feature selection method by using the adaptive graph to preserve the local manifold structure. [55] extended this graph learning strategy to multiview data. [59] proposed to use Eq. (2) to construct Anchor graph for large-scale hyperspectral image clustering. There are also many other applications of Eq. (2) that are beyond the scope of this paper.

However, one big issue is never noticed: it is highly sensitive to noisy input data. This is obvious in Eq. (2) since S is built from the raw data X, which is often corrupted in practice. Consequently, the performance of this approach is deteriorated. To remedy this, we present a principle to robustify graph learning.

3 Robust Graph Construction

3.1 Formulation

To make the graph learning robust and reliable, we introduce our approach in this section. The core idea is that the observed data matrix is not perfect but corrupted by errors. Thus, instead of operating on original data or performing some data cleaning that precedes the analysis, we assume the data to be decomposed into two parts: the clean data D and the corruptions E. Then, the graph learning is performed on clean data. Following the RPCA idea that the clean data is low-rank and corruptions are sparse [56,60,61], we jointly learn the clean data and the graph. Finally, our proposed Robust Graph Construction (RGC) model can be formulated as:

$$\min_{D,E,S} \|D\|_* + \alpha\|E\|_1 + \beta Tr(DLD^T) + \gamma\|S\|_F^2,$$

$$s.t. \ X = D + E, \ S\mathbf{1} = \mathbf{1}, \ 0 \le S \le 1, \tag{3}$$

4

where α, β, and γ are all trade-off parameters. We will show later that γ can be computed based on a reasonable assumption. In the proposed formulation Eq. (3), graph learning and noise removal are implemented together, such that they can be iteratively boosted.

If we assume that the graph S is artificially computed from original data X, then L is directly available. Finally, S and L are no longer unknowns. Problem (3) changes to:

$$\min_{D,E} \|D\|_* + \alpha\|E\|_1 + \beta Tr(DLD^T),$$
$$s.t. \ X = D + E. \tag{4}$$

This is nothing but the RPCA model, except that it incorporates the manifold smoothness regularizer term. It assumes that the low-rank data D itself lies on a smooth manifold. This is called MRPCA model [62].

To summarize, compared with existing work in the literature, our proposed objective function in Eq. (3) enjoys the following advantages:

- The graph construction is implemented on clean data. Consequently, the graph quality would be improved. This would be helpful for clustering and semisupervised classification.

- Interestingly, Eq. (3) can also be treated as an RPCA model. Different from popular RPCA models, it imposes the smooth manifold regularizer. As a result, the low-rank recovery would be enhanced.

- Furthermore, not like the existing graph Laplacian construction method, our graph Laplacian is adaptively learned from raw data. Thus, it would be robust to noise and outliers.

3.2 Optimization Algorithm

To solve Eq. (3), we first introduce auxiliary variable Z to facilitate the solution of D. Then Eq. (3) can be written in the following equivalent form:

$$\min_{D,E,S,Z} \|D\|_* + \alpha\|E\|_1 + \beta Tr(ZLZ^T) + \gamma\|S\|_F^2,$$
$$s.t. \ X = D + E, \ S\mathbf{1} = \mathbf{1}, \ 0 \le S \le 1, \ Z = D. \tag{5}$$

It can be solved via alternating direction method of multipliers (ADMM). Removing the equality constraints on X and Z, we obtain the augmented Lagrangian function as follows:

$$\mathcal{L}(D,E,S,Z,Y_1,Y_2) = \|D\|_* + \alpha\|E\|_1 + \beta Tr(ZLZ^T) +$$
$$\gamma\|S\|_F^2 + \frac{\mu}{2}(\|D + E - X + \frac{Y_1}{\mu}\|_F^2 + \|D - Z + \frac{Y_2}{\mu}\|_F^2), \tag{6}$$
$$s.t. \ S\mathbf{1} = \mathbf{1}, \ 0 \le S \le 1,$$

5

where $\mu > 0$ is a penalty parameter and Y_1, Y_2 are Lagrangian multipliers. We can solve those unknown variables alternatingly, one at each step, while keeping the others fixed.

Step 1: Updating D while fixing other variables. The problem Eq. (6) becomes:

$$\min_{D} \|D\|_* + \mu\|D - H\|_F^2, \tag{7}$$

where $H = \frac{X+Z-E-(Y_1+Y_2)/\mu}{2}$. It has closed-form solution according to singular value shrinkage, i.e., $D = U diag((\sigma - \frac{1}{2\mu})_+)V^T$, $U diag(\sigma)V^T$ is the singular value decomposition (SVD) of H.

Step 2: Updating E. We have

$$\min_{E} \alpha\|E\|_1 + \frac{\mu}{2}(\|E - (X - D - \frac{Y_1}{\mu})\|_F^2. \tag{8}$$

It also admits close-formed solution, i.e., $e_{ij} = (|g_{ij}| - \alpha/\mu)_+ \cdot sign(g_{ij})$, where $G = X - D - \frac{Y_1}{\mu}$.

Step 3: Updating S. Remember that L is also a function of S, so the problem Eq. (6) becomes:

$$\min_{s_i} \sum_{j=1}^{n}(\frac{\beta}{2}\|z_i - z_j\|^2 s_{ij} + \gamma s_{ij}^2), \tag{9}$$

$$s.t. \ s_i^T \mathbf{1} = 1, \ 0 \le s_{ij} \le 1.$$

Denote $f_{ij} = \|z_i - z_j\|^2$ and $f_i \in \mathcal{R}^{n \times 1}$, then the above subproblem can be reformulated as:

$$\min_{s_i^T \mathbf{1}=1, 0 \le s_{ij} \le 1} \|s_i + \frac{\beta}{4\gamma}f_i\|^2 \tag{10}$$

The problem naturally has a sparse solution. Therefore, we just update its k nearest neighbors, i.e., s_i has k positive entries and $s_{ij} = 0$ for $j > k$. The Lagrangian function of it is:

$$\mathcal{L}(s_i, \eta, \xi) = \|s_i + \frac{\beta}{4\gamma_i}f_i\|^2 - \eta(s_i^T \mathbf{1} - 1) - \xi_i^T s_i, \tag{11}$$

where η and $\xi \in \mathcal{R}^{n \times 1}$ are the Lagrangian multipliers and the overall γ can be set to the average of $\{\gamma_i\}_{i=1}^{n}$. By the Karush-Kuhn-Tucker condition, it yields $s_i = (\frac{\eta}{2} - \frac{\beta f_i}{4\gamma_i})_+$.

Let's rank f_i in ascending order, we obtain

$$\begin{cases} s_{ik} = \frac{\eta}{2} - \frac{\beta f_{ik}}{4\gamma_i} > 0 \\ s_{i,k+1} = \frac{\eta}{2} - \frac{\beta f_{i,k+1}}{4\gamma_i} \le 0 \\ s_i^T \mathbf{1} = \sum_{j=1}^{k}(\frac{\eta}{2} - \frac{\beta f_{ij}}{4\gamma_i}) = 1 \end{cases} \Rightarrow \begin{cases} s_{ij} = \frac{f_{i,k+1} - f_{ij}}{k f_{i,k+1} - \sum_{r=1}^{k} f_{ir}}, \ j \le k \\ \gamma_i = \frac{\beta}{4}(k f_{i,k+1} - \sum_{j=1}^{k} f_{ij}) \\ \eta = \frac{2}{k} + \frac{\beta}{2k\gamma_i} \sum_{j=1}^{k} f_{ij} \end{cases} \tag{12}$$

In the above derivations, we set the value of γ_i to its maximum. Then, we take the average of $\{\gamma_i\}_{i=1}^n$, we have

$$\gamma = \frac{\beta}{4n} \sum_{i=1}^{n} (k f_{i,k+1} - \sum_{j=1}^{k} f_{ij}).\tag{13}$$

This ensures that the average number of nonzero elements in each row of S is close to k.

Step 4: Updating Z. We have

$$\min_{Z} \beta Tr(ZLZ^T) + \frac{\mu}{2} \|D - Z + \frac{Y_2}{\mu}\|_F^2).\tag{14}$$

Its first-order derivative is $2\beta ZL - \mu(D - Z + \frac{Y_2}{\mu})$. By setting it to zero, we achieve

$$Z = (\mu D + Y_2)(2\beta L + \mu I)^{-1}.\tag{15}$$

The details of the algorithm are given in Algorithm 1. As we can see, the main computation load of this algorithm is from SVD involved in Eq. (7) and matrix inversion in Eq. (15). Their computational complexity is $O(n^3)$ in general. Some existing packages can be employed to speed up their computation, e.g., partial SVD.

Algorithm 1 The Proposed Method RGC

Input: Data matrix X, parameters $\alpha > 0$, $\beta > 0$, γ, $\mu > 0$.
Initialize: $Z = X$, $E = 0$, $Y_1 = Y_2 = 0$.
While not converge **do**
 1: Calculate D by (7).
 2: Update E according to (8).
 3: Update S using (12)
 4: Calculate Z using (15).
 5: Update Lagrange multipliers Y_1 and Y_2 as

$$Y_1 = Y_1 + \mu(D + E - X),$$
$$Y_2 = Y_2 + \mu(D - Z).$$

End while.

---

4   Experiments on Clustering

In this section, we demonstrate the effectiveness of our robust graph construction technique towards clustering task. We compare it with several representative methods on a number of benchmark data sets.

Table 1: Description of the data sets

	# instances	# features	# classes
YALE	165	1024	15
JAFFE	213	676	10
ORL	400	1024	40
COIL20	1440	1024	20
BA	1404	320	36
TR11	414	6429	9
TR41	878	7454	10
TR45	690	8261	10

4.1 Comparison Methods

- **Spectral Clustering (SC)** [29]. SC is a widely used clustering technique. It can separate different shaped groups. However, how to build a good graph is still a challenge. As we show later, we manually construct 12 graphs. For our proposed RGC method, we obtain clustering results by performing spectral clustering with our learned S.

- **Robust Kernel K-means (RKKM)** [22]. As an extension to classical k-means clustering method, RKKM has the capability of dealing with non-linear structures, noise, and outliers in the data. RKKM shows promising results on a number of real-world data sets.

- **Constructing Robust Affinity Graph for Spectral Clustering (RSC)** [63]. In contrast to employing all features to construct a graph, RSC builds the graph via identifying and exploiting discriminative features by using random forest approach.

- **Simplex Sparse Representation (SSR)** [44]. Based on the self-expressiveness property, SSR constructs a sparse graph for clustering. It achieves satisfying performance in numerous data sets.

- **Clustering with Adaptive Neighbor (CAN)** [51]. Based on the idea of adaptive neighbor, CAN learns a graph from raw data for clustering task. Different from CAN, our method learns the graph from clean data.

- **Twin Learning for Similarity and Clustering (TLSC)** [45]: Based on self-expressiveness, TLSC has been proposed recently and has shown superior performance on a number of real-world data sets. TLSC learns graph matrix in kernel space. Since kernels are built from the original data, this method is also sensitive to noise.

- **Manifold RPCA (MRPCA)** [62]. Rather than learn the graph from clean data, MRPCA constructs the graph from raw data according to some metrics. K-means is implemented on D to achieve cluster labels.

- Our proposed **RGC**[1]. We learn reliable graphs from raw data, which is often corrupted in reality.

(a) Accuracy(%)

Data	RKKM [22]	SC [29]	RSC [63]	SSR [44]	CAN [51]	TLSC [45]	MRPCA [62]	RGC
YALE	48.09	49.42	57.58	54.55	58.79	55.85	62.42	**64.85**
JAFFE	75.61	74.88	98.59	87.32	98.12	**99.83**	98.12	98.59
ORL	54.96	57.96	62.75	69.00	61.50	62.35	56.75	**73.00**
COIL20	61.64	67.60	72.99	76.32	84.58	72.71	73.47	**85.42**
BA	42.17	31.07	44.01	23.97	36.82	47.72	44.16	**51.00**
TR11	53.03	50.98	34.54	41.06	38.89	71.26	33.82	**71.37**
TR41	56.76	63.52	60.93	63.78	62.87	67.43	34.97	**70.16**
TR45	58.13	57.39	48.41	71.45	56.96	74.02	31.88	**75.51**

(b) NMI(%)

Data	RKKM [22]	SC [29]	RSC [63]	SSR [44]	CAN [51]	TLSC [45]	MRPCA [62]	RGC
YALE	52.29	52.92	60.25	57.26	57.67	56.50	62.86	**65.29**
JAFFE	83.47	82.08	98.16	92.93	97.31	**99.35**	97.72	98.13
ORL	74.23	75.16	79.87	84.23	76.59	78.96	76.11	**84.35**
COIL20	74.63	80.98	83.91	86.89	91.55	82.20	83.73	**91.58**
BA	57.82	50.76	58.17	30.29	49.32	63.04	57.57	**64.89**
TR11	49.69	43.11	24.77	27.60	19.17	58.60	8.10	**67.74**
TR41	60.77	61.33	56.78	59.56	51.13	65.50	26.50	**67.35**
TR45	57.86	48.03	43.70	67.82	49.31	**74.24**	17.81	71.74

(c) Purity(%)

Data	RKKM [22]	SC [29]	RSC [63]	SSR [44]	CAN [51]	TLSC [45]	MRPCA [62]	RGC
YALE	49.79	51.61	63.03	58.18	59.39	56.50	57.27	**66.48**
JAFFE	71.82	76.83	97.65	96.24	98.12	**99.85**	98.12	98.13
ORL	59.60	61.45	79.87	76.50	68.50	74.00	62.50	**76.75**
COIL20	66.35	69.92	80.83	89.03	87.85	82.53	80.28	**90.90**
BA	45.28	45.28	41.95	40.85	39.67	52.36	50.29	**57.69**
TR11	67.93	67.93	35.75	85.02	44.20	**82.85**	62.32	79.85
TR41	74.99	74.99	55.58	75.40	67.54	73.23	41.80	**74.03**
TR45	68.18	68.18	45.51	83.62	60.87	**78.26**	55.65	77.39

Table 2: Clustering results measured on benchmark data sets. The best results for these algorithms are highlighted in boldface.

---

[1] https://github.com/sckangz/RGC

4.2 Data Sets

To fully examine the performance of our algorithm, we implement clustering experiments on eight publicly available data sets. We summarize the statistics information in Table 1. Specifically, YALE, JAFFE, and ORL are face databases, COIL20 is a toy image database, and BA is a binary alpha digits data set. They are taken under different illumination conditions or with different configurations, so some of them are corrupted severely. The last three data sets are text corpora derived from TERC[2].

4.3 Evaluation Metrics

Three popular metrics are used to quantitatively assess our algorithm's performance on clustering task. They are accuracy (Acc), normalized mutual information (NMI), and Purity.

Acc measures the one-to-one relationship between clusters and classes. Let l_i and $\hat{l}_i$ be the algorithm's output and the ground truth label of x_i, respectively. Then the Acc can be defined as

$$Acc = \frac{\sum_{i=1}^{n} \delta(\hat{l}_i, map(l_i))}{n},$$

where n denotes the number of samples, $\delta(\cdot)$ is the delta function, and $map(\cdot)$ is the best permutation mapping function. Based on Kuhn-Munkres algorithm, $map(\cdot)$ maps each cluster index to a true class label based.

The NMI is designed to measure the quality of clustering. Let L and $\hat{L}$ denote two sets of clusters, then their marginal probability distribution functions can be represented by $p(l)$ and $p(\hat{l})$, respectively, induced from the joint distribution $p(l, \hat{l})$. Finally, the NMI can be defined by

$$\text{NMI}(L, \hat{L}) = \frac{\sum\limits_{l \in L, \hat{l} \in \hat{L}} p(l, \hat{l}) \log(\frac{p(l, \hat{l})}{p(l)p(\hat{l})})}{\max(H(L), H(\hat{L}))},$$

The greater NMI means the better clustering performance.

The last evaluation metric that we adopt is the purity, which evaluates the extent to which the most common category in each cluster. It is computed as follows:

$$\text{Purity} = \sum_{i=1}^{c} \frac{n_i}{n} P(C_i), \quad P(S_i) = \frac{1}{n_i} max_j(n_i^j),$$

where n_i is the number of points in cluster C_i, and n_i^j represents the total number of points that the i-th input group is assigned to the j-th category. There are c categories in total. It is easy to see that a larger Purity indicated better clustering performance.

4.4 Setup

For RKKM, SC, TLSC, MRPCA methods, we prepare their inputs by following the setup in [22]. In particular, we design 12 kernels. They are: seven Gaussian kernels of the form $K(x, y) = exp(-\|x - y\|_2^2/(td_{max}^2))$, where d_{max} is the maximal distance between samples and t varies over the set $\{0.01, 0.0, 0.1, 1, 10, 50, 100\}$; a linear kernel $K(x, y) = x^T y$; four polynomial kernels $K(x, y) = (a + x^T y)^b$ with $a \in \{0, 1\}$ and $b \in \{2, 4\}$. Besides, all kernels are rescaled to $[0, 1]$ by dividing each element by the largest pairwise squared distance. Then we report their best performances in Table 2.

There are three parameters in our model. According to RPCA, we set $\alpha = 1/\sqrt{max(m, n)}$ for all our experiments. And γ is computed by (13) in every iteration. In all experiments, we set neighbor number $k = 10$. As a result, the only unknown parameter is β, and for this we can use cross validation.

4.5 Results

We evaluate the clustering performance in terms of three widely used metrics: accuracy (Acc), normalized mutual information (NMI), and Purity. As can be seen from Table 2, our method outperforms other state-of-the-art techniques in most experiments. Specifically, we have the following observations: 1) Our proposed method can beat CAN in all cases. Note that RGC constructs graph from clean data, while CAN learns graph from raw data. In particular, RGC improves the Acc of CAN from 36.83%, 38.89%, 56.96% to 51%, 71.37%, 75.51% on BA, TR41, TR45, respectively. This fully demonstrates the importance of constructing a robust graph; 2) Compared with self-expressiveness based graph construction methods SSR and TLSC, our method RGC achieves better performance in most cases. However, the performance of CAN is not as good as SSR and TLSC in most experiments. This also verifies the importance of removing noise in the data; 3) RGC performs much better than MRPCA in all cases. This is attributed to our adaptive graph learning strategy; 4) Compared with RKKM and SC, we also observe considerable improvements, although they have tried different kinds of kernels and RKKM is supposed to cope with noise; 5) With respect to RSC, RGC also has big advantages in terms of Acc, NMI and Purity. To summarize, our proposed graph construction method can improve clustering performance significantly w.r.t. several newly developed techniques, this is mainly due to the removal of noise and errors.

4.6 Parameter Analysis

As discussed in subsection 4.4, model parameter β and auxiliary parameter μ are unknown in our algorithm. Take YALE data set as an example, we show the sensitivity of our performance to their values. We can observe that Acc, NMI, Purity behave similarly with varying β and μ. Our method works well for a wide range of β and μ values. Moreover, we show the progress of objective function values of Eq. (3) in Figure 1. It is observed that our algorithm converges very
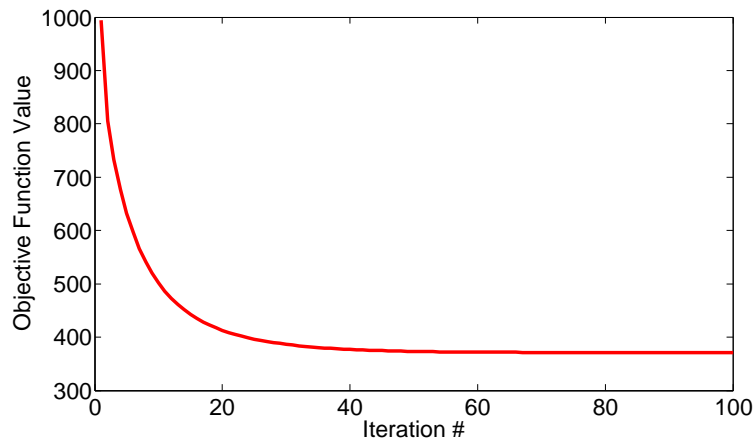
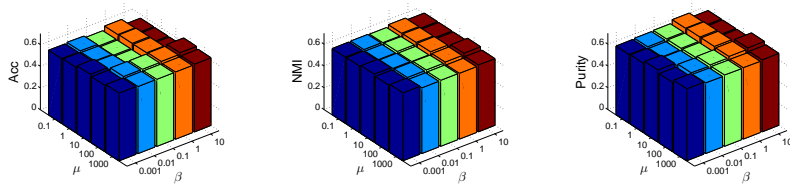Figure 1: Convergence curve of the objective function value (3).

Figure 2: The influence of β and μ values on accuracy, NMI, and Purity of YALE data set.

fast.

5 Semisupervised Classification

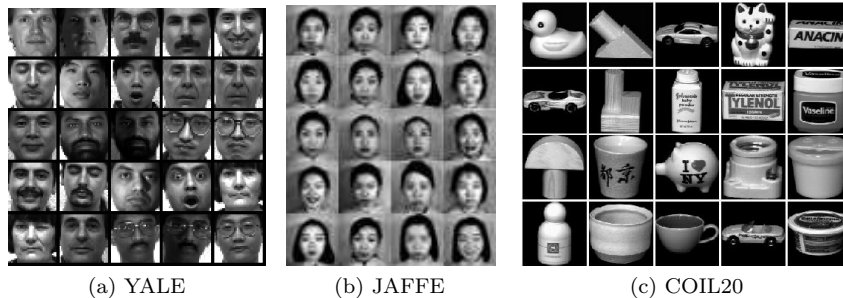(a) YALE   (b) JAFFE   (c) COIL20

Figure 3: Sample images of YALE, JAFFE, and COIL20.

In this section, we show that our method also performs well on semisupervised classification task. We first obtain the graph matrix S from RGC, then we use the popular local and global consistency (LGC) method to perform classification task [64]. Specifically, LGC achieves a classification function $F \in \mathcal{R}^{n \times c}$ by solving the following problem:

$$\min_{F} Tr\{F^T L F + \gamma (F - Y)^T (F - Y)\}, \tag{16}$$

where c is the class number, $Y \in \mathcal{R}^{n \times c}$ is the label matrix, in which $y_{ij} = 1$ iff the i-th sample belongs to the j-th class, and $y_{ij} = 0$ otherwise. As before, L is the graph Laplacian matrix calculated from S.

5.1 Data Sets

We evaluate our method on two types of recognition tasks: face recognition and visual object recognition. In specific, we examine the effectiveness of our robust graph learning for face recognition on frequently used YALE and JEFFE data sets. The YALE face database has 15 individuals, and each individual has 11 near frontal images which are taken under different lighting conditions. Figure 3a displays some sample photos. The JAFFE face data set contains 10 individuals, and each person shows 7 different facial expressions, including 6 basic facial expressions and 1 neutral. We use COIL20 data set to perform visual object recognition experiment. It contains 20 objects and there are 72 photos for each object. For this data set, the images are taken 5 degrees apart when the object is rotating on a turntable. Figure 3c shows some images from this database.

5.2 Comparison Algorithms

We compare RGC with five existing methods.

- **Local and Global Consistency (LGC)** [64]: LGC is a widely used semisupervised classification method.

- **Gaussian Field and Harmonic function (GFHF)** [65]: Besides LGC, GFHF is another popular label propagating technique.

- **Semisupervised Classification with Adaptive Neighbors (SCAN)** [55]: This recently developed method uses adaptive neighbors approach to construct the similarity graph. Moreover, the graph construction and clustering are formulated into a unified framework to improve the performance.

- **Self-expression graph based Semisupervised Learning** [66]: Different from SCAN, Li et al. propose to learn a graph for semisupervised learning by using self-expression method. Similar to SCAN, the similarity matrix and class indicator matrix F are updated alternatingly, so they can help improve each other. Based on the low-rank and sparse assumption of the graph matrix, there are two models: S^2LRR and S^3R.

Among these comparison techniques, both LGC and GFHF take L as input. To obtain better performance, we calculate L based on 7 kernel matrices. Then we report the best results from them. In specific, four Gaussian kernels with $t \in \{0.1, 1, 10, 100\}$; a linear kernel $K(x, y) = x^y$; two polynomial kernels of the form with $K(x, y) = (a + x^T y)^2$ $a \in \{0, 1\}$. For other techniques, the graph is learned from data.

Table 3: Classification accuracy (%) on benchmark data sets (mean±standard deviation). The best results are in bold font.

Data	Labeled Percentage(%)	GFHF [65]	LGC [64]	S^3R [66]	S^2LRR [66]	SCAN [55]	RGC
YALE	10	38.00±11.91	47.33±13.96	38.83±8.60	28.77±9.59	45.07±1.30	**58.90**±13.44
	30	54.13±9.47	63.08±2.20	58.25±4.25	42.58±5.93	60.92±4.03	**69.13**±3.84
	50	60.28±5.16	69.56±5.42	69.00±6.57	51.22±6.78	68.94±4.57	**69.83**±5.95
JAFFE	10	92.85±7.76	96.68±2.76	97.33±1.51	94.38±6.23	96.92±1.68	**98.81**±0.72
	30	98.50±1.01	98.86±1.14	99.25±0.81	98.82±1.05	98.20±1.22	**99.31**±0.81
	50	98.94±1.11	99.29±0.94	99.82±0.60	99.47±0.59	99.25±5.79	**99.87**±0.59
COIL20	10	87.74±2.26	85.43±1.40	93.57±1.59	81.10±1.69	90.09±1.15	**95.71**±1.04
	30	95.48±1.40	87.82±1.03	96.52±0.68	87.69±1.39	95.27±0.93	**97.33**±0.92
	50	98.62±0.71	88.47±0.45	97.87±0.10	90.92±1.19	97.53±0.82	**99.80**±0.18
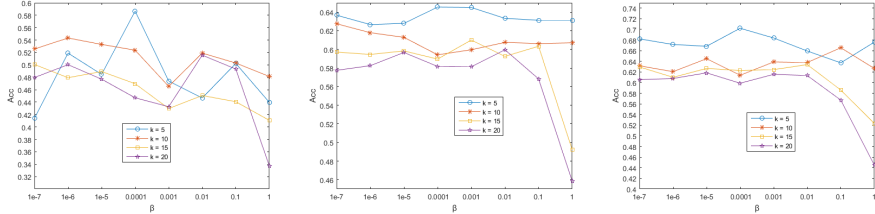
(a) Labeled Percentage (10%) (b) Labeled Percentage (30%) (c) Labeled Percentage (50%)

Figure 4: The influence of β and neighbor number k on YALE dataset.

5.3 Results

To see the effect of label number, we randomly choose 10%, 30%, 50% of samples as labeled. We repeat this process 20 times and report the average classification accuracy and deviation in Table 3. Unsurprisingly, the accuracy of all techniques goes up when the number of labeled samples increases. This is consistent with our intuition. As can be seen, our proposed RGC consistently outperforms other existing techniques. This demonstrates the importance of noise removal. Concretely, RGC improves on LGC significantly verifies the importance of graph construction. Please notice that LGC directly adopts manually built graph, while our method uses the learned graph S. Therefore, the enhancements are resulted from our high-quality graph construction. Another interesting observation about RGC is that the improvements are more considerable when the labeling ratio is relatively low. This makes RGC more attracting in real applications where labeled instances are extremely rare.

Though graph construction and label learning are implemented in a unified framework in SCAN, S²LRR, and S³R, their performance is still not as good as our two-step approach. Once again, this is attribute to our graph learning strategy. Since SCAN and RGC are both based on adaptive neighbors graph construction, the superiority of RGC over SCAN is due to that RGC learns graph from clean data.

5.4 Parameter Sensitivity

We illustrate the parameter sensitivity for semisupervised learning task on YALE dataset in Figure 4. It is important to note that $k = 5$ often produces the highest accuracy. Therefore, the results reported in Table 2 and 3, where $k = 10$ is adopted, could be further improved. Besides, our algorithm provides reasonable results for a wide range of β value.

15

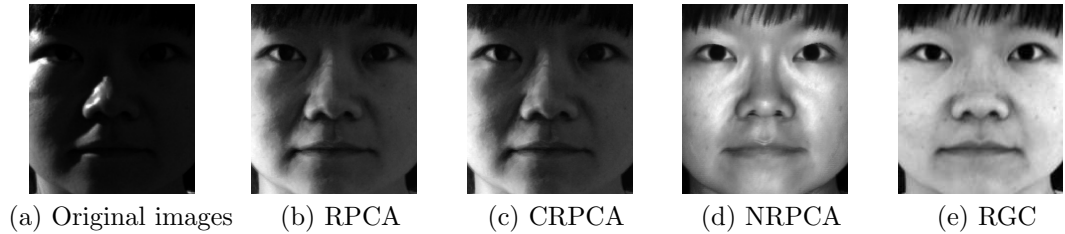(a) Original images (b) RPCA (c) CRPCA (d) NRPCA (e) RGC

Figure 5: Shadow removal from face images. Column 1 displays sample images. Columns 2 to 5 show the recovered results obtained by different algorithms.

6 Experiments on Data Recovery

In this section, we assess the performance of our model (3) in low-rank matrix recovery task. RPCA has numerous applications. Due to space limitation, we focus on two practical tasks: shadow removal from face images and background extraction from videos.

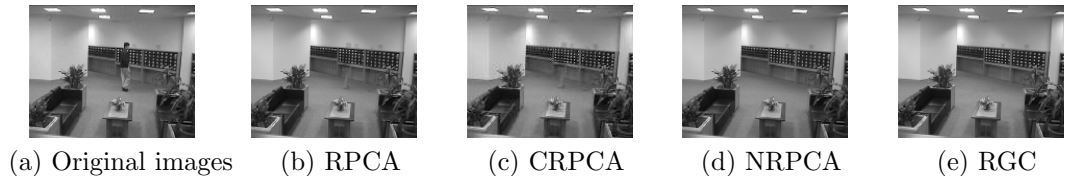(a) Original images (b) RPCA (c) CRPCA (d) NRPCA (e) RGC

Figure 6: Low-rank background recovery. Column 1 displays actual frame. Columns 2 to 5 show the recovered low-rank background using different algorithms.

6.1 Comparison Methods

We choose a good set of methods to compare with our proposed method.

- **Robust Principle Component Analysis (RPCA)** [56]. Recovering clean data with low-rank structure from the corrupted data has been intensively studied during the last decade. One of the most representative techniques is RPCA. Low-rank part and sparse part can effectively characterize the clean data and errors, respectively.

- **Capped-norm based RPCA (CRPCA)** [67]. RPCA uses the nuclear norm to approximate the rank function, which generally leads to an NP-hard problem. However, the nuclear norm shrinks all the singular values equally and thus over-penalizes large singular values, which often results in a biased solution. CRPCA adopts a truncation strategy to alleviate the nuclear norm deficiency and leads to better performance.

- **Nonconvex RPCA (NRPCA)** [68]. By combining the simplicity of PCA and elegant theory of RPCA, NRPCA has been developed recently. It alternatingly projects the residuals onto the low-rank and sparse matrix sets. It has global convergence and exact recovery guarantee. However, it also requires the knowledge of sparsity,

- Our proposed **RGC** method. Compared with RPCA, our method incorporates manifold smoothness effect. Both CRPCA and NRPCA adopt a nonconvex approach, while our method still uses convex nuclear norm.

6.2 Face Image Shadow Removal

Removing shadows, specularities, and saturations in face images is crucial to face recognition. With a bunch of face images, we are supposed to recover the clean image. We randomly choose one subject from the Extended Yale B database. There are 64 images of size 192×168 taken under different lighting conditions. All images are vectorized, hence $X \in \mathcal{R}^{32256 \times 64}$.

Figure 5 shows the recovered images from an example image. We can observe that our proposed RGC totally removes the specularities and shadows. And there are still some artifacts left for other three methods, especially for RPCA and CRPCA.

6.3 Background Extraction from Videos

An important application of low-rank recovery is to separate static background from the dynamic foreground. We use benchmark data set lobby, which contains 1,546 frames of size 128×160. After vectorizing and concatenating, the size of X becomes $20,480 \times 1,546$. Since the low-rank ground truth is not available, we present a visual comparison of recovered low-rank background in Figure 6. We can see that our method totally removes the walking person, while some shadows of pant are left in the frame recovered by RPCA, so the presence of graph in our model enables better recovery. For NRPCA, if we assume a small rank value (e.g., 2 here), we can also get a pretty clean frame.

7 Conclusion

In this paper, we present a model to learn reliable graphs from raw data. It aims to address the robustness issue of adaptive neighbor graph learning method. From another point of view, it is a generalization of the Robust PCA by leveraging graph regularization on the low-rank representation. Consequently, our proposed framework can either enhance clustering/semisupervised classification or improve low-rank recovery from grossly corrupted data sets. Experiments on several benchmark data sets reveal that our method outperforms various state-of-the-art clustering/semisupervised classification and data recovery techniques.

References

[1] C. Peng, Z. Kang, Y. Hu, J. Cheng, and Q. Cheng, "Nonnegative matrix factorization with integrated graph and feature learning," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 8, no. 3, p. 42, 2017.

[2] Z. Kang, C. Peng, Q. Cheng, and Z. Xu, "Unified spectral clustering with optimal graph," in *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence (AAAI-18)*, 2018.

[3] S. Huang, Z. Kang, I. W. Tsang, and Z. Xu, "Auto-weighted multi-view clustering via kernelized graph learning," *Pattern Recognition*, 2018.

[4] Z. Kang, Y. Lu, Y. Su, C. Li, and Z. Xu, "Similarity learning via kernel preserving embedding," in *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence (AAAI-19)*, 2019.

[5] C. Peng, Z. Kang, and Q. Cheng, "Integrating feature and graph learning with low-rank representation," *Neurocomputing*, vol. 249, pp. 106–116, 2017.

[6] X. Chen, Y. Ye, X. Xu, and J. Z. Huang, "A feature group weighting method for subspace clustering of high-dimensional data," *Pattern Recognition*, vol. 45, no. 1, pp. 434–446, 2012.

[7] Y. Yang, F. Shen, Z. Huang, H. T. Shen, and X. Li, "Discrete nonnegative spectral clustering," *IEEE Transactions on Knowledge and Data Engineering*, vol. 29, no. 9, pp. 1834–1845, 2017.

[8] Z. Wang, Q. Ling, and T. Huang, "Learning deep l0 encoders," in *AAAI Conference on Artificial Intelligence*, 2016, pp. 2194–2200.

[9] X. Chang, F. Nie, Z. Ma, Y. Yang, and X. Zhou, "A convex formulation for spectral shrunk clustering." in *AAAI*, 2015, pp. 2532–2538.

[10] S. Huang, Z. Kang, and Z. Xu, "Self-weighted multi-view clustering with soft capped norm," *Knowledge-Based Systems*, 2018.

[11] Y. Ren, C. Domeniconi, G. Zhang, and G. Yu, "Weighted-object ensemble clustering: methods and analysis," *Knowledge and Information Systems*, vol. 51, no. 2, pp. 661–689, 2017.

[12] F. Wang, T. Li, G. Wang, and C. Zhang, "Semi-supervised classification using local and global regularization." in *AAAI*, vol. 8, 2008, pp. 726–731.

[13] Y.-F. Li, S.-B. Wang, and Z.-H. Zhou, "Graph quality judgement: a large margin expedition," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*. AAAI Press, 2016, pp. 1725–1731.

[14] L. Berton and A. De Andrade Lopes, "Graph construction for semi-supervised learning," in *Proceedings of the 24th International Conference on Artificial Intelligence*, ser. IJCAI'15. AAAI Press, 2015, pp. 4343–4344.

[15] F. Nie, J. Li, X. Li *et al.*, "Parameter-free auto-weighted multiple graph learning: A framework for multiview clustering and semi-supervised classification." in *IJCAI*, 2016, pp. 1881–1887.

[16] G. Chao and S. Sun, "Semi-supervised multi-view maximum entropy discrimination with expectation laplacian regularization," *Information Fusion*, vol. 45, pp. 296–306, 2019.

[17] A. Li, Z. Wu, H. Lu, D. Chen, and G. Sun, "Collaborative self-regression method with nonlinear feature based on multi-task learning for image classification," *IEEE Access*, vol. 6, pp. 43 513–43 525, 2018.

[18] C. Peng, Z. Kang, and Q. Cheng, "Subspace clustering via variance regularized ridge regression," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, HI, USA*, 2017, pp. 21–26.

[19] Y. Ren, K. Hu, X. Dai, L. Pan, S. C. Hoi, and Z. Xu, "Semi-supervised deep embedded clustering," *Neurocomputing*, vol. 325, pp. 121–130, 2019. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0925231218312049

[20] X. Chen, W. Hong, F. Nie, D. He, M. Yang, and J. Z. Huang, "Directly minimizing normalized cut for large scale data," in *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, KDD-18*, 2018, pp. 1206–1215.

[21] S. Huang, Y. Ren, and Z. Xu, "Robust multi-view data clustering with multi-view capped-norm k-means," *Neurocomputing*, 2018.

[22] L. Du, P. Zhou, L. Shi, H. Wang, M. Fan, W. Wang, and Y.-D. Shen, "Robust multiple kernel k-means using $\ell_{2,1}$-norm," in *Proceedings of the 24th International Conference on Artificial Intelligence*. AAAI Press, 2015, pp. 3476–3482.

[23] X. Liu, L. Wang, J. Yin, E. Zhu, and J. Zhang, "An efficient approach to integrating radius information into multiple kernel learning," *IEEE Trans. Cybernetics*, vol. 43, no. 2, pp. 557–569, 2013.

[24] X. Chen, X. Xu, Y. Ye, and J. Z. Huang, "TW-k-means: Automated Two-level Variable Weighting Clustering Algorithm for Multi-view Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 25, no. 4, pp. 932–944, 2013.

[25] H. Liu, J. Wu, T. Liu, D. Tao, and Y. Fu, "Spectral ensemble clustering via weighted k-means: Theoretical and practical evidence," *IEEE transactions on knowledge and data engineering*, vol. 29, no. 5, pp. 1129–1143, 2017.

[26] C. Peng, Z. Kang, S. Cai, and Q. Cheng, "Integrate and conquer: Double-sided two-dimensional k-means via integrating of projection and manifold construction," *ACM Transactions on Intelligent Systems and Technology (TIST)*, vol. 9, no. 5, p. 57, 2018.

[27] X. Liu, X. Zhu, M. Li, L. Wang, C. Tang, J. Yin, D. Shen, H. Wang, and W. Gao, "Late fusion incomplete multi-view clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, pp. 1–1, 2018.

[28] U. Von Luxburg, "A tutorial on spectral clustering," *Statistics and computing*, vol. 17, no. 4, pp. 395–416, 2007.

[29] A. Y. Ng, M. I. Jordan, Y. Weiss *et al.*, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 2, pp. 849–856, 2002.

[30] Y. Wang, L. Wu, X. Lin, and J. Gao, "Multiview spectral clustering via structured low-rank matrix factorization," *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[31] Z. Kang, C. Peng, and Q. Cheng, "Kernel-driven similarity learning," *Neurocomputing*, vol. 267, pp. 210–219, 2017.

[32] Y. Yang, Z. Ma, Y. Yang, F. Nie, and H. T. Shen, "Multitask spectral clustering by exploring intertask correlation," *IEEE transactions on cybernetics*, vol. 45, no. 5, pp. 1083–1094, 2015.

[33] J. Yu, D. Tao, M. Wang, and Y. Rui, "Learning to rank using user clicks and visual features for image retrieval," *IEEE transactions on cybernetics*, vol. 45, no. 4, pp. 767–779, 2015.

[34] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering." in *NIPS*, vol. 17, no. 1601-1608, 2004, p. 16.

[35] H. Zhu, R. Vial, S. Lu, X. Peng, H. Fu, Y. Tian, and X. Cao, "Yotube: Searching action proposal via recurrent and static regression networks," *IEEE Transactions on Image Processing*, vol. 27, no. 6, pp. 2609–2622, 2018.

[36] K. Zhan, C. Zhang, J. Guan, and J. Wang, "Graph learning for multiview clustering," *IEEE transactions on cybernetics*, 2017.

[37] S. Li, H. Liu, Z. Tao, and Y. Fu, "Multi-view graph learning with adaptive label propagation," in *Big Data (Big Data), 2017 IEEE International Conference on*. IEEE, 2017, pp. 110–115.

[38] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the l2-graph for robust subspace learning and subspace clustering," *IEEE transactions on cybernetics*, vol. 47, no. 4, pp. 1053–1066, 2017.

[39] H. Zhu, X. Peng, V. Chandrasekhar, L. Li, and J.-H. Lim, "Dehazegan: When image dehazing meets differential programming." in *IJCAI*, 2018, pp. 1234–1240.

[40] A. Li, D. Chen, Z. Wu, G. Sun, and K. Lin, "Self-supervised sparse coding scheme for image classification based on low rank representation," *PloS one*, vol. 13, no. 6, p. e0199141, 2018.

[41] M. Zhao, T. W. Chow, Z. Zhang, and B. Li, "Automatic image annotation via compact graph based semi-supervised learning," *Knowledge-Based Systems*, vol. 76, pp. 148–165, 2015.

[42] Z. Kang, L. Wen, W. Chen, and Z. Xu, "Low-rank kernel learning for graph-based clustering," *Knowledge-Based Systems*, vol. 163, pp. 510–517, 2019.

[43] X. Peng, C. Lu, Y. Zhang, and H. Tang, "Connections between nuclear norm and frobenius norm based representation," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 29, no. 1, pp. 218–224, Jan. 2018.

[44] J. Huang, F. Nie, and H. Huang, "A new simplex sparse learning model to measure data similarity for clustering." in *IJCAI*, 2015, pp. 3569–3575.

[45] Z. Kang, C. Peng, and 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.

[46] G. Liu, Z. Lin, S. Yan, J. Sun, Y. Yu, and Y. Ma, "Robust recovery of subspace structures by low-rank representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 35, no. 1, pp. 171–184, 2013.

[47] E. Elhamifar and R. Vidal, "Sparse subspace clustering: Algorithm, theory, and applications," *IEEE transactions on pattern analysis and machine intelligence*, vol. 35, no. 11, pp. 2765–2781, 2013.

[48] X. Peng, J. Feng, S. Xiao, W. Y. Yau, J. T. Zhou, and S. Yang, "Structured autoencoders for subspace clustering," *IEEE Trans Image Process*, vol. 27, no. 10, pp. 5076–5086, Oct 2018.

[49] L. Zhuang, H. Gao, Z. Lin, Y. Ma, X. Zhang, and N. Yu, "Non-negative low rank and sparse graph for semi-supervised learning," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*. IEEE, 2012, pp. 2328–2335.

[50] Z. Kang, X. Lu, J. Yi, and Z. Xu, "Self-weighted multiple kernel learning for graph-based clustering and semi-supervised classification," in *IJCAI*, 2018, pp. 2312–2318.

[51] F. Nie, X. Wang, and H. Huang, "Clustering and projected clustering with adaptive neighbors," in *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2014, pp. 977–986.

[52] L. Zhang, Q. Zhang, B. Du, J. You, and D. Tao, "Adaptive manifold regularized matrix factorization for data clustering," in *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, 2017, pp. 3399–3405.

[53] L. Du and Y.-D. Shen, "Unsupervised feature selection with adaptive structure learning," in *Proceedings of the 21th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. ACM, 2015, pp. 209–218.

[54] X. Liu, L. Wang, J. Zhang, J. Yin, and H. Liu, "Global and local structure preservation for feature selection," *IEEE Trans. Neural Netw. Learning Syst.*, vol. 25, no. 6, pp. 1083–1095, 2014.

[55] F. Nie, G. Cai, and X. Li, "Multi-view clustering and semi-supervised classification with adaptive neighbours." in *AAAI*, 2017, pp. 2408–2414.

[56] E. J. Candès, X. Li, Y. Ma, and J. Wright, "Robust principal component analysis?" *Journal of the ACM (JACM)*, vol. 58, no. 3, p. 11, 2011.

[57] X. He and P. Niyogi, "Locality preserving projections," in *Advances in neural information processing systems*, 2004, pp. 153–160.

[58] S. Cai, Z. Kang, M. Yang, X. Xiong, C. Peng, and M. Xiao, "Image denoising via improved dictionary learning with global structure and local similarity preservations," *Symmetry*, vol. 10, no. 5, p. 167, 2018.

[59] R. Wang, F. Nie, and W. Yu, "Fast spectral clustering with anchor graph for large hyperspectral images," *IEEE Geoscience and Remote Sensing Letters*, vol. 14, no. 11, pp. 2003–2007, 2017.

[60] Z. Kang, C. Peng, and Q. Cheng, "Robust pca via nonconvex rank approximation," in *Proceedings of the 2015 IEEE International Conference on Data Mining (ICDM)*. IEEE Computer Society, 2015, pp. 211–220.

[61] M. Yang, W. Li, and M. Xiao, "On identifiability of higher order block term tensor decompositions of rank rank-1," *Linear and Multilinear Algebra*, pp. 1–23, 2018.

[62] N. Shahid, V. Kalofolias, X. Bresson, M. Bronstein, and P. Vandergheynst, "Robust principal component analysis on graphs," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2812–2820.

[63] X. Zhu, C. Change Loy, and S. Gong, "Constructing robust affinity graphs for spectral clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014, pp. 1450–1457.

[64] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Advances in neural information processing systems*, 2004, pp. 321–328.

[65] X. Zhu, Z. Ghahramani, and J. D. Lafferty, "Semi-supervised learning using gaussian fields and harmonic functions," in *Proceedings of the 20th International conference on Machine learning (ICML-03)*, 2003, pp. 912–919.

[66] C.-G. Li, Z. Lin, H. Zhang, and J. Guo, "Learning semi-supervised representation towards a unified optimization framework for semi-supervised learning," in *Proceedings of the IEEE International Conference on Computer Vision*, 2015, pp. 2767–2775.

[67] Q. Sun, S. Xiang, and J. Ye, "Robust principal component analysis via capped norms," in *Proceedings of the 19th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2013, pp. 311–319.

[68] P. Netrapalli, U. Niranjan, S. Sanghavi, A. Anandkumar, and P. Jain, "Non-convex robust pca," in *Advances in Neural Information Processing Systems*, 2014, pp. 1107–1115.