

Received April 7, 2020, accepted April 22, 2020, date of publication April 27, 2020, date of current version May 14, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.2990562

Robust Similarity-Based Concept Factorization for Data Representation

XINGYU SHEN^{1,2}, XIANG ZHANG^{1,3}, LONG LAN^{1,3}, (Member, IEEE),
QING LIAO⁴, (Member, IEEE), AND ZHIGANG LUO^{1,2}, (Member, IEEE)

¹Science and Technology on Parallel and Distributed Processing Laboratory, National University of Defense Technology, Changsha 410073, China

²College of Computer, National University of Defense Technology, Changsha 410073, China

³State Key Laboratory of High Performance Computing, Institute of Quantum Information, National University of Defense Technology, Changsha 410073, China

⁴Department of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), Shenzhen 518055, China

Corresponding authors: Xiang Zhang (zhangxiang08@nudt.edu.cn), Long Lan (long.lan@nudt.edu.cn), and Zhigang Luo (zgluo@nudt.edu.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61806213.

ABSTRACT Non-negative matrix factorization (NMF) known as learnt parts-based representation has become a data analysis tool for clustering tasks. It provides an alternative learning paradigm to cope with non-negative data clustering. In this paradigm, concept factorization (CF) and symmetric non-negative factorization (SymNMF) are two typically important representative models. In general, they have distinct behaviors: in CF, each cluster is modeled as a linear combination of samples, and vice versa, i.e., sample reconstruction, while SymNMF built on pair-wise sample similarity measure, is to preserve similarity of samples in a low-dimensional subspace, namely similarity reconstruction. In this paper, we propose a similarity-based concept factorization (SCF) as a synthesis of the two behaviors. This design can be formulated as: the similarity of reconstructed samples by CF is close to that of original samples. To optimize it, we develop an optimization algorithm which leverages the alternating direction of multipliers (ADMM) method to solve each sub-problem of SCF. Besides, we take a further step to consider the robust issue of similarity reconstruction and explore a robust SCF model (RSCF), which penalizes the hardest pair-wise similarity reconstruction via l_∞ . Thus, RSCF enjoys *similarity preservation*, *robustness to similarity perturbation*, and *ability of reconstructing samples*. Extensive experiments validate such properties and show that the proposed SCF and RSCF achieve large performance gains as compared to their counterparts.

INDEX TERMS Non-negative matrix factorization, concept factorization, l_∞ norm, the optimal gradient method, sorting-based algorithm.

I. INTRODUCTION

Non-negative matrix factorization (NMF, [1], [2]) approximates a non-negative data matrix by a product of two low-rank non-negative matrices, namely, the basis matrix and the coefficient matrix. Of them, the coefficient matrix provides an effective way to represent original high-dimensional data. More importantly, NMF is known as learning parts-based representation, i.e., the parts composed of the whole, which is line with human intuition.

Building upon this intuition, NMF has become an important data analysis tool for various tasks such as clustering tasks [3], [4], recommendation systems [5], [6],

speech separation [7]. Specifically, in theory, a square loss NMF is also equals to K -means [8], which is a classical clustering method. Thus, the learnt basis matrix could be interpreted as the clusters while the coefficient matrix indicates the cluster identity of each sample. Obviously, NMF is a linear method based on original feature space. To avoid this issue, many nonlinear methods such as kernel NMF [9] have been explored. A representative method is concept factorization (CF, [10]), which assumes that each cluster is modeled as a linear combination of samples, and each sample is modeled as a linear combination of clusters. The assumption can be called sample reconstruction. Besides, the similarity measures are critical for many clustering tasks, such as community detection [11], person re-identification and image retrieval. A typical similarity-based method is symmetric

The associate editor coordinating the review of this manuscript and approving it for publication was Hao Luo.

NMF (SymNMF, [12]), which is on the basis of similarity measurement to preserve the similarity among samples via the so-called similarity reconstruction. That is, the coefficients learned in the low-dimensional subspace have the same similarity relationships among original samples. For sake of this character, SymNMF has achieved sound clustering performance. Kang *et al.* [13] proposed a self-tuning similarity learning method relying on the kernel-based embedding, called SLKE. They argued that similarity preserving could help the new representation relay on the overall relations among the data. SLKE could get well clustering performance. SLKE and SymNMF emphasize similarity information while CF has the ability to reconstruct data. That inspires us whether we can combine the advantages of the two ideas to achieve higher clustering performance. Thus, this work tries to a synthesis of such two ideas by jointly integrating both sample reconstruction and similarity reconstruction into a uniform formulation.

Different from SymNMF, which shows the potential of similarity preservation, graph-based NMF methods regard the manifold structure within dataset as the regularizer to achieve the identical goal. For instance, graph regularized NMF (GNMF, [14]) utilizes the Laplacian regularization term to smooth the learnt low-dimensional representation. According to [15], LCCF implements a locally invariant CF method. LCCF seems the joint product of both the similarity preservation and sample reconstruction. In fact, the graph regularizer behaves as the sample smooth rather than similarity reconstruction. Thus, GNMF and LCCF are different from our goal.

Besides the mentioned issues, another key focus is the robust issue. In general, most existing works focus on sample reconstruction distortion in the case samples are corrupted with the noises [16]. To name a few, Kong *et al.* assumed the noises obeying the Laplacian distribution thus equipping NMF with the L_{21} loss and yielding robust NMF namely L_{21} NMF [17]; Liutkus *et al.* thought the noise subject to isotropic Cauchy distribution and thus developed CauchyNMF [18]; based on Gaussian kernelization correntropy induced metric, Du *et al.* proposed CIMNMF [19]. Distinct from them, we study the robust issue about the similarity reconstruction distortion.

To consider the aforementioned insights, this paper proposes a similarity-based CF method (SCF) and its robust variant (RSCF). In particular, SCF intends to reconstruct the similarity of the reconstructed data meanwhile preserve the representation of original data information. This provides a synthesis of two behaviors, i.e., sample reconstruction and similarity reconstruction. Fig. 1 and Fig. 2 show the ability that the similarity of reconstructed samples by SCF is close to that of original samples. To achieve the robust issue about similarity measurement, we introduce l_∞ norm instead of L_2 norm to measure the distortion levels of similarity reconstruction errors, which considers the similarity reconstruction in the worst-case assumption. This way complements existing robust NMF methods. To solve both methods, we develop an

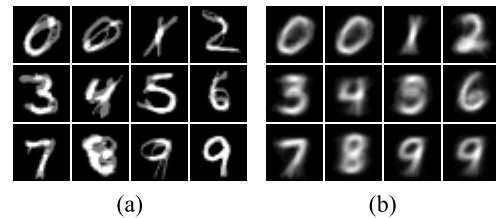


FIGURE 1. The original images (a) and the corresponding reconstructed images by SCF (b) from MNIST dataset. The original images (a) are randomly selected from the test set.

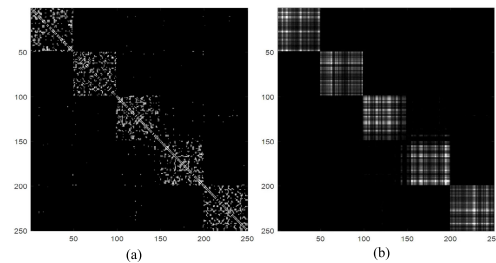


FIGURE 2. The original similarity (a) and the corresponding reconstructed similarity by SCF (b) from MNIST dataset. We randomly select 50 images of top five classes from the test set and calculate the original similarity matrix (a) of them.

efficient algorithm under the ADMM framework. Clustering experiments on several popular datasets show the efficacy of our proposed methods.

To sum up, this study has the following contributions:

- 1) A concept factorization method based on similarity reconstruction is proposed, called SCF. It reconstructs the pairwise similarity to obtain better data presentation.
- 2) We incorporate the l_∞ norm with proposed SCF to enhance the robustness to tiny distortions of similarity reconstruction.
- 3) Since the objective functions of SCF and RSCF are fourth-order functions which is non-convex, we develop an optimization algorithm based on ADMM to solve them.

The rest parts fall into four sections as below. In section II, we have a brief review to the most related work including NMF, CF, robust method, and l_∞ norm. Our proposed methods and the corresponding optimization algorithms are presented in the section III. Section IV details clustering experiments conducted on several facial datasets. In the final, we draw a conclusion in section V.

II. RELATED WORK

A. NON-NEGATIVE MATRIX FACTORIZATION

NMF is a classical method for clustering which uses a non-negative constraint purely additive perception process to characterize how data consists of parts. NMF is in efforts to factorize the high-dimensional data by the linear representation of clusters. Lee and Seung [1] utilized Euclidean distance

to measure the quality of reconstruction. A standard NMF can be formed as following:

$$\min_{U, V \geq 0} \|X - UV\|_F^2 \quad (1)$$

where $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$ is the non-negative input data and each sample is $x_i \in \mathbb{R}^{d \times 1}$ ($i = 1, 2, 3, \dots, n$). $U \in \mathbb{R}^{d \times r}$ and $V \in \mathbb{R}^{r \times n}$ are two low rank matrices. The optimization problem of Eq. (1) is non-convex for joint U and V which is difficult to obtain a global optimal solution using a nonlinear optimization method. However, for sub-problem about U or V only, it is still a convex optimization problem (i.e., it is convex when one variable is fixed and another variable is optimized). Lee and Seung used the multiplicative update rule (MUR, [2]) to update U and V alternately. The nonincreasing update rules are:

$$U_{ij} = U_{ij} \times \frac{(XV^T)_{ij}}{(UVV^T)_{ij}}, \quad (2)$$

$$V_{ij} = V_{ij} \times \frac{(U^T X)_{ij}}{(U^T UV)_{ij}}. \quad (3)$$

It is shown that the update rules can ensure the convergence of the algorithm. More detail can be found in [20].

B. CONCEPT FACTORIZATION

Xu and Gong thought the non-negative matrix factorization method is only suitable for original features [10]. For highly non-linearly distributed data sets, the NMF does not perform well while the kernel method can be used to improve the matrix decomposition results [21]. The proposed method called concept factorization (CF) can apply kernel method to matrix factorization. The CF considered the basis matrix as a linear combination of data and the objective function can be shown as:

$$\min_{W, H \geq 0} \|X - XWH\|_F^2. \quad (4)$$

If we regard XW as a whole, it can be interpreted as the clustering center of clustering task in the objective function Eq. (4) while H is the corresponding indicator matrix. With the analogous MUR algorithm [22], the corresponding update criterion of variable is given as

$$W_{ab} = W_{ab} \times \frac{(KH^T)_{ab}}{(KWHH^T)_{ab}}, \quad (5)$$

$$H_{ab} = H_{ab} \times \frac{(W^T K)_{ab}}{(W^T KWH)_{ab}}, \quad (6)$$

in which $K = X^T X$.

C. l_∞ OPTIMIZATION

l_∞ is utilized in our methods thus we review it here. l_∞ function measures the maximum fitting errors rather than a square loss which can help yield the global minimum rather than

a local one. Thus, l_∞ is very useful for geometric problems. Indeed, l_∞ minimization problem is to measure worst-case distortion. Aghazadeh *et al.* [23] argued that l_∞ is more robust to small perturbations. They designed a data-dependent hash function for retrieval and achieve the state-of-art performance. Richard Hartley and Frederik Schaffalitzky claim that l_∞ cost function is significantly simpler than the l_2 cost [24]. They found that l_∞ minimization optimization can come down to minimizing a cost function with a single minimum (local or global) on a convex domain. The analysis on multi-view triangulation and motion recovery problems demonstrates that the optimal problems of small dimension solution are reliable.

Assume that z is a signal, an efficient representation x can be solved by [25]:

$$\arg \min_x \|x\|_\infty \quad \text{subject to } \|Ax - z\| \leq \varepsilon, \quad (7)$$

in which A is transformation matrix and ε is the error tolerance. The minimization problem of Eq. (7) can be arisen as signal processing, wireless communications and various areas. If C is designed as a domain $\{v | \|v\| < \varepsilon\}$, a constrained form of Eq. (7) is transformed as

$$\arg \min_{x, c \in C} \|x\|_\infty \quad \text{subject to } c = \|Ax - z\|. \quad (8)$$

After that, we enforce the equality constraint introducing Lagrange multiplier φ and yield a saddle-point formulation problem:

$$\arg \min_{x, c \in C} \|x\|_\infty + \langle \varphi, Ax - z - c \rangle, \quad (9)$$

where $\langle a, b \rangle = a^T b$. The Primal-Dual Hybrid Gradient (PDHG) method is a powerful optimization scheme [26] that breaks complex problems into simple sub-steps to solve the problem in Eq. (9). Before applying PDGH to the optimization problem, we need to evaluate a proximal minimization first:

$$\arg \min_{x, c \in C} \|x\|_\infty + \frac{1}{2\kappa} \|x - \tilde{x}\|, \quad (10)$$

where κ is a positive penalty parameter. A low computation cost described in [27] can perform the proximal operator update to solve Eq. (10).

D. ROBUST METHODS

In the real world, the data cannot be in a flawless case that is pure or has slight noise. Some unwished outliers always affect the model performance even inducing unreasonable results. Assume the given data goes with an unknown corruption matrix, we take the NMF as an example and obtain an object function with corrupted data [28]:

$$\begin{aligned} \|(X + \xi)_{ij} - (UV)_{ij}\|_F^2 &= \|E + \xi\|_F^2 \\ &= \sum_{j,k} (E_{jk}^2 + 2\xi_{jk}E_{jk} + \xi_{jk}^2), \end{aligned} \quad (11)$$

where $E = X - UV$ is the reconstruction error matrix. If the ξ is tiny even 0, the limited impact may be neglect. However, an overrun ξ can severely impact the performance even cause disastrous consequences. To descent this deficiency, many robust methods are proposed to increase the immunity to outliers. Many robust methods are based on a hypothesis that the corruption data is sampled from some distributions. For example, $L_{2,1}$ NMF [17] considered that the corruption existed in data space followed the Laplace distribution and $L_{2,1}$ norm could respond it effectively. CIMNMF [19] based on the correntropy induced metric applied a Gaussian kernel method. Cauchy NMF [18] employed the isotropic Cauchy distribution to overcome the sensitivity of outliers. Recently, Shen *et al.* explored a non-statistic-based method called tanhNMF [28]. They rethink a hyperbolic tangent (*tanh*) function to evaluate the reconstruction error. It is also an approximate truncation method. All above are devoting to seek a proper metric to reduce the abnormal data sensitivity.

III. PROPOSED METHOD

This section introduces our similarity-based concept factorization (SCF) and its robust version based similarity construction (RSCF). Besides, we provide an efficient optimization algorithm based on ADMM to solve them.

A. SCF

As discussed above, CF is to model each cluster as a linear combination of samples and each sample as a linear combination of cluster centers. This implies the following data reconstruction assumption:

$$\Phi(X) \approx \Phi(X)WH, \quad (12)$$

where Φ is an implicit nonlinear mapping function which maps original samples into a infinite-dimensional feature space in the RKHS. The objective function of CF is formulated as:

$$\min_{W, H \geq 0} \|\Phi(X) - \Phi(X)WH\|_F^2. \quad (13)$$

As one knows, similarity preservation or manifold structure within dataset plays a core role in clustering tasks [29]. The representative methods include spectral clustering [30] and symmetric NMF (SymNMF, [12]). Of them, SymNMF minimizes similarity reconstruction error to learn cluster identities, and outperforms spectral clustering in most situations. In fact, SymNMF implies that the similarity of the learnt representation in a low-dimensional subspace should approach to that of original samples. Thus, the insight of similarity reconstruction can be represented as:

$$K \approx H^T H, \quad (14)$$

where $K \in \mathbb{R}^{n \times n}$ is the similarity matrix of n samples. The construction of K is an important aspect of affecting the model performance. In general, a good similarity can greatly

improve the effectiveness of the model. This scope is not our focus, thus we do not plan to detail how to build a “perfect” similarity. In our model, we use classical k nearest neighbor to metric the similarity [31].

According to two distinct behaviors as above, i.e., data reconstruction and similarity reconstruction, can we couple them each other into a unified reconstruction error like either Eq. (12) or Eq. (14)? The answer is yes! We couple them with the assumption that the similarity of the reconstructed samples is close to that of original samples. The insight can be written as:

$$K = \Phi(X)^T \Phi(X) \\ \approx (\Phi(X)WH)^T (\Phi(X)WH) = H^T W^T KWH. \quad (15)$$

In terms of Eq. (15), the reconstruction error is evaluated by the Frobenius norm, and we yield our similarity-based concept factorization (SCF) as follows:

$$F = \min_{W, H \in \mathbb{R}_+} \|K - H^T W^T KWH\|_F^2. \quad (16)$$

From Eq. (16), the equivalent form is recast as:

$$\|\Phi(X)^T \Phi(X) - (\Phi(X)WH)^T (\Phi(X)WH)\|_F^2 \\ = \|(\Phi(X) - \Phi(X)WH)(\Phi(X) + \Phi(X)WH)\|_F^2. \quad (17)$$

The minimization of Eq. (17) implies that the matrix $(\Phi(X) - \Phi(X)WH)(\Phi(X) + \Phi(X)WH)$ tends to zero. Due to the non-negativity of all the factors including $\Phi(X)$, W and H , $\Phi(X) + \Phi(X)WH$ is always positive and cannot be a zero matrix. According to algebra theory, to make Eq. (17) to be zero, the $\Phi(X) - \Phi(X)WH$ could be zero or belong to the null space of $\Phi(X) + \Phi(X)WH$. Interestingly, Eq. (17) expands the solution space, which might contain much more useful information for clustering tasks. This character might be the reason for the effectiveness of SCF.

B. OPTIMIZATION ALGORITHM

Obviously, the optimization problems of Eq. (16) on W and H are non-convex owing to their fourth-order terms. Empirically, using MUR to optimize Eq. (16) will fail to converge. To overcome the difficulty, we introduce two variables to reduce Eq. (16) to several optimization sub-problems, and then utilize the ADMM based algorithm [32] to solve them.

By introducing two variables J and M , the optimization problem Eq. (16) becomes:

$$\begin{aligned} & \text{minimize } \|K - J^T KM\|_F^2 \\ & \text{subject to } J = WH \\ & \quad M = WH \\ & \quad W > 0, H > 0. \end{aligned} \quad (18)$$

We can write the augmented Lagrangian function [33] of this optimization problem as:

$$\begin{aligned} L(W, H, J, M, Y_1, Y_2) \\ = & \|K - J^T K M\|_F^2 \\ & + \sum_i^n \left[Y_{1i}^T ((WH)_i - J_i) + \frac{\lambda}{2} \|(WH)_i - J_i\|_2^2 \right] \\ & + \sum_i^m \left[Y_{2i}^T ((WH)_i - M_i) + \frac{\gamma}{2} \|(WH)_i - M_i\|_2^2 \right], \quad (19) \end{aligned}$$

where Y_1 and Y_2 signify the Lagrange multipliers, and both λ and γ are the positive penalty parameters. According to **Corollary 1**, the optimization sub-problems with respect to each variable are convex.

Corollary 1: Eq. (19) is a convex sub-optimization problem about W, H, J, M respectively, which can be written as follows:

$$L(W, H) = \frac{\lambda}{2} \|WH - J + \frac{Y_1}{\lambda}\|_F^2 + \frac{\gamma}{2} \|WH - M + \frac{Y_2}{\gamma}\|_F^2, \quad (20)$$

$$\begin{aligned} L(J, K) = & \|K - J^T K M\|_F^2 + \frac{\lambda}{2} \|WH - J + \frac{Y_1}{\lambda}\|_F^2 \\ & + \frac{\gamma}{2} \|WH - M + \frac{Y_2}{\gamma}\|_F^2. \quad (21) \end{aligned}$$

The proof can be found in **Appendix A**.

We now describe the specific optimization process for each variable.

1) OPTIMIZATION W AND H

Corollary 2: According to Eq. (20), the gradient of $L(W, H)$ with respect to W and H is Lipschitz continuous with corresponding Lipschitz constant $P_W = (\lambda + \gamma) \|HH^T\|_2$ and $P_H = (\lambda + \gamma) \|W^T W\|_2$, respectively.

We deduce the proof of **Corollary 2** in **Appendix B**. According to **Corollary 1** and **Corollary 2**, we can use the optimal gradient method (OGM, [34]–[36]) to solve them. In summary, we sum up the whole optimization procedure of W and H into **Algorithm 1** and **Algorithm 2**, respectively.

2) OPTIMIZATION J AND M

According to Eq. (21), it is easy to find that the optimization problem about J is a quadratic minimization problem, which usually has an analytic solution. First, we can get the corresponding partial derivative:

$$\begin{aligned} \frac{\partial L(J, M)}{\partial J} &= \frac{\partial \text{Tr}(-2M^T K J K + M^T K J J^T K M)}{\partial J} \\ &\quad + \frac{\lambda}{2} \frac{\partial \text{Tr}(J^T J - 2J^T (WH + \frac{Y_1}{\lambda}))}{\partial J} \\ &= -2KMK + 2KMM^T KJ + \lambda J - \lambda WH - Y. \quad (22) \end{aligned}$$

Algorithm 1 Nesterov's Optimal Gradient Method (OGM) for W

Input:

A nonnegative matrix $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$, H^k , W^k , Lagrangian multipliers Y_1^k and Y_2^k after k iterations, positive penalty parameters λ and γ , the stopping criterions η , *maxiter*;

Output:

The basis matrix W ;

- 1: Initialize $P_W = (\lambda + \gamma) \|H^k (H^k)^T\|_2$, $\Gamma_1 = W^k$, $\alpha_1 = 1$, iteration counter $t = 1$, $\varepsilon = \infty$;
- 2: **while** $\varepsilon > \eta$ or $t < \text{maxiter}$ **do**
- 3: $W_{t+1} = \max \left(0, \Gamma_t - \frac{1}{P_W} \nabla_W L' \right)$;
- 4: $\alpha_{t+1} = \frac{1 + \sqrt{1 + 4\alpha_t^2}}{2}$;
- 5: $\Gamma_t = W_{t+1} + \frac{\alpha_t - 1}{\alpha_{t+1}} (W_{t+1} - W_t)$;
- 6: $t \leftarrow t + 1$;
- 7: $\varepsilon = \frac{\|W_{t+1} - W_t\|_F^2}{\|W_t\|_F^2}$;
- 8: **end while**
- 9: $W = W_{t+1}$;
- 10: **return** W_{t+1} .

Algorithm 2 Nesterov's Optimal Gradient Method (OGM) for H

Input:

A nonnegative matrix $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$, H^k , W^k , Lagrangian multipliers Y_1^k and Y_2^k after k iterations, positive penalty parameters λ and γ , the stopping criterions η , *maxiter*;

Output:

The indicator matrix H ;

- 1: Initialize $P_H = (\lambda + \gamma) \|(W^k)^T W^k\|_2$, $\Gamma_1 = H^k$, $\alpha_1 = 1$, iteration counter $t = 1$, $\varepsilon = \infty$;
- 2: **while** $\varepsilon > \eta$ or $t < \text{maxiter}$ **do**
- 3: $H_{t+1} = \max \left(0, \Gamma_t - \frac{1}{P_H} \nabla_H L' \right)$;
- 4: $\alpha_{t+1} = \frac{1 + \sqrt{1 + 4\alpha_t^2}}{2}$;
- 5: $\Gamma_t = H_{t+1} + \frac{\alpha_t - 1}{\alpha_{t+1}} (H_{t+1} - H_t)$;
- 6: $t \leftarrow t + 1$;
- 7: $\varepsilon = \frac{\|H_{t+1} - H_t\|_F^2}{\|H_t\|_F^2}$;
- 8: **end while**
- 9: $H = H_{t+1}$;
- 10: **return** W_{t+1} .

By setting the partial derivative to 0, we can obtain:

$$\begin{aligned} -2KMK + 2KMM^T KJ + \lambda J - \lambda WH - Y &= 0 \\ \Leftrightarrow J &= \left(2KMM^T K + \lambda I \right)^{-1} (2KMK + \lambda WH + Y). \quad (23) \end{aligned}$$

where I is the identity matrix and $(\cdot)^{-1}$ represents the inverse of matrix.

Analogously, the derivative for M is

$$\frac{\partial L(J, M)}{\partial M} = -2JK + 2JJ^T M + \lambda M - \lambda WH - Y_2. \quad (24)$$

Then, we have the following closed-form solution:

$$M = (2KJJ^T KM + \gamma I)^{-1} (KJK + \gamma WH + Y_2). \quad (25)$$

3) OPTIMIZE LAGRANGIAN MULTIPLIERS Y_1 AND Y_2

The Lagrangian multipliers in ADMM can be updated by dual ascent methods. Thus, Y_1 and Y_2 have the following two update procedures:

$$Y_1 = Y_1 + \lambda (WH - J), \quad (26)$$

$$Y_2 = Y_2 + \gamma (WH - M). \quad (27)$$

During the optimization process, the two penalty parameters often constantly increase to the infinity. The generally-used approach is to multiply by a constant $s > 1$. In this paper, λ and γ have $\lambda \leftarrow s\lambda$ and $\gamma \leftarrow s\gamma$, respectively.

The whole optimization for our SCF is summarized into **Algorithm 3**. We must be honest with a truth that **Algorithm 3** cannot guarantee that the objective function is monotonically non-increasing at each iteration round. The main reason is that the Lagrangian multiplier increases, which causes the objective function to oscillate.

Algorithm 3 ADMM Based Algorithm for SCF

Input:

non-negative matrix $X = [x_1, x_2, \dots, x_n] \in \mathbb{R}^{d \times n}$,
positive penalty parameters λ and γ , increment step s ;

Output:

A basis matrix W and an indicator matrix H ;

- 1: Initialize $Y_1 = \mathbf{0}_{n \times n}$, $Y_2 = \mathbf{0}_{n \times n}$, $J = \mathbf{I}_{n \times n}$,
 $M = \mathbf{I}_{n \times n}$, $k = 0$, randomly initialize W^k, H^k ;
 - 2: **while** not convergent **do**
 - 3: $W^{k+1} = OGM_W(W^k, H^k, J^k, M^k, Y_1^k, Y_2^k, \lambda, \gamma)$;
 - 4: $H^{k+1} = OGM_H(W^{k+1}, H^k, J^k, M^k, Y_1^k, Y_2^k, \lambda, \gamma)$;
 - 5: $J^{k+1} = \arg \min_J L(W^{k+1}, H^{k+1}, J^k, M^k, Y_1^k, Y_2^k, \lambda, \gamma)$;
 - 6: $M^{k+1} = \arg \min_M L(W^{k+1}, H^{k+1}, J^{k+1}, M^k, Y_1^k, Y_2^k, \lambda, \gamma)$;
 - 7: $Y_1^{k+1} = Y_1^k + \lambda (W^{k+1} H^{k+1} - J^{k+1})$;
 - 8: $Y_2^{k+1} = Y_2^k + \gamma (W^{k+1} H^{k+1} - M^{k+1})$;
 - 9: $k = k + 1$;
 - 10: $\lambda \leftarrow s\lambda$;
 - 11: $\gamma \leftarrow s\gamma$;
 - 12: **end while**
 - 13: $W = W_{t+1}$; $H = H_{t+1}$;
 - 14: **return** W, H .
-

C. ROBUST SCF

In SCF, the similarity matrix is built on the k nearest neighbors, thus it is easily influenced by the noisy corruptions in

real world. To be special, the paired similar samples could have small similarity value. In general, these values violate the whole data distribution. Thus, we recover the similarity values in the worst-case situation.

Given Z is the gap between the similarity of original samples and reconstructed samples, that is, $Z = K - H^T W^T KWH$. Different from Eq. (16), minimizing the worst-case reconstruction error implies to minimize the maximum reconstruction distortions. Thus, we rewrite Eq. (16) as:

$$\begin{aligned} & \text{minimize } \sup_i |Z_{ij}| \\ & \text{subject to } Z = K - J^T KM. \end{aligned} \quad (28)$$

Eq. (28) caters to the Chebyshev norm [37], i.e., the l_∞ norm. As usual, the l_∞ norm of a matrix is different from that of a vector. Then, the matrix Z can be reshaped as a long vector $\tilde{Z}$. To this end, the objective function Eq. (28) can be reformulated as:

$$\begin{aligned} & \text{minimize } \|\tilde{Z}\|_\infty \\ & \text{subject to } Z = K - J^T KM, \end{aligned} \quad (29)$$

where $\|\cdot\|_\infty$ is the above Chebyshev norm.

To optimize Eq. (29), we introduce another two auxiliary variables to obtain the following problem:

$$\begin{aligned} F = \min_{W, H \geq 0} & \|\tilde{Z}\|_\infty \\ & \text{subject to } Z = K - J^T KM \\ & J = WH \\ & M = WH \\ & W > 0, H > 0. \end{aligned} \quad (30)$$

We adopt the ADMM framework like SCF to solve the optimization problem Eq. (30). We enforce the equality constraints by introducing three Lagrange multipliers Λ_0 , Λ_1 and Λ_2 . The augmented Lagrangian function can be recast as:

$$\begin{aligned} F(Z, W, H, J, M, \Lambda_0, \Lambda_1, \Lambda_2) &= \|\tilde{Z}\|_\infty \\ &+ \sum_i \left[\Lambda_{0i}^T \left(Z_i - (K - J^T KM)_i \right) \right] \\ &+ \sum_i \left[\frac{\rho}{2} \|Z_i - (K - J^T KM)_i\|_2^2 \right] \\ &+ \sum_i \left(\Lambda_{1i}^T ((WH)_i - J_i) + \frac{\lambda}{2} \|(WH)_i - J_i\|_2^2 \right) \\ &+ \sum_i \left(\Lambda_{2i}^T ((WH)_i - M_i) + \frac{\gamma}{2} \|(WH)_i - M_i\|_2^2 \right), \end{aligned} \quad (31)$$

where ρ , λ and γ are the positive penalty parameters which also increase constantly with a step s . The update procedures are described as follows.

1) OPTIMIZATION ON Z

Fixed other variables, the optimization problem corresponding to Z can be evaluated by a proximal problem of l_∞ norm which can be written as:

$$\begin{aligned} F(Z) &= \|\tilde{Z}\|_\infty + \sum_i \left[\Lambda_{0i}^T \left(Z_i - \left(K - J^T KM \right)_i \right) \right] \\ &\quad + \sum_i \left[\frac{\rho}{2} \left\| Z_i - \left(K - J^T KM \right)_i \right\|_2^2 \right] \\ &= \|\tilde{Z}\|_\infty + \frac{\rho}{2} \left\| Z - \left(K - J^T KM \right) + \frac{\Lambda_0}{\rho} \right\|_F^2 - \frac{\|\Lambda_0\|_F^2}{\rho}. \end{aligned} \quad (32)$$

For the purpose of optimizing Eq. (31), we consider reshape K, J, M and Λ_0 to their column vector representation $\tilde{K}, \tilde{J}, \tilde{M}$ and $\tilde{\Lambda}_0$, respectively. An equivalent optimization can be written as:

$$\begin{aligned} \min_z \|\tilde{Z}\|_\infty + \frac{\rho}{2} \left\| Z - \left(K - J^T KM \right) + \frac{\Lambda_0}{\rho} \right\|_F^2 \\ \Leftrightarrow \min_z \|\tilde{Z}\|_\infty + \frac{\rho}{2} \left\| \tilde{Z} - \left(\tilde{K} - \tilde{J}^T \tilde{K} \tilde{M} \right) + \frac{\tilde{\Lambda}_0}{\rho} \right\|_2^2. \end{aligned} \quad (33)$$

We obtain the saddle point of Eq. (33) by using the proximal optimization scheme based on a sorting algorithm. After the sub-problem is optimized, $\tilde{Z}$ is restored to the original matrix with the same size as Z . We use ' $\odot$ ' signifies the elementwise product and the sorting-based algorithm for optimization is summarized in **Algorithm 3**.

2) OPTIMIZATION ON J AND M

We can also write the sub-problems with respect to J and M as:

$$\begin{aligned} F(J, M) &= \frac{\rho}{2} \left\| Z - \left(K - J^T KM \right) + \frac{\Lambda_0}{\rho} \right\|_F^2 \\ &\quad + \frac{\lambda}{2} \left\| WH - J + \frac{\Lambda_1}{\lambda} \right\|_F^2 + \frac{\gamma}{2} \left\| WH - M + \frac{\Lambda_2}{\gamma} \right\|_F^2. \end{aligned} \quad (34)$$

We can get the gradients of them as:

$$\begin{aligned} \frac{\partial L}{\partial J} &= \rho KM \left(Z - K + \frac{\Lambda_0}{\rho} \right)^T + \rho KMM^T K^T J \\ &\quad + \lambda J - \lambda WH - \Lambda_1, \end{aligned} \quad (35)$$

$$\begin{aligned} \frac{\partial L}{\partial M} &= \rho KJ \left(Z - K + \frac{\Lambda_0}{\rho} \right) + \rho KJJ^T K^T M \\ &\quad + \gamma M - \gamma WH - \Lambda_2, \end{aligned} \quad (36)$$

and then, we set the Eq. (35) and Eq. (36) to 0. The close solutions can be obtained:

$$\begin{aligned} J &= \left(\rho KMM^T K^T + \lambda I \right)^{-1} \\ &\quad \times \left(\lambda WH + \Lambda_1 - \rho KM \left(Z - K + \frac{\Lambda_0}{\rho} \right)^T \right), \end{aligned} \quad (37)$$

$$\begin{aligned} M &= \left(\rho KJJ^T K^T + \gamma I \right)^{-1} \\ &\quad \times \left(\gamma WH + \Lambda_2 - \rho KJ \left(Z - K + \frac{\Lambda_0}{\rho} \right) \right). \end{aligned} \quad (38)$$

In summary, the whole procedure of optimizing RSCF is detailed in **Algorithm 4**.

Algorithm 4 Sorting-Based Algorithm(SA) for Z **Input:**

Similarity matrix K , auxiliary matrix J, M , the lagrangian multiplier Λ_0 and positive penalty parameter ρ

Output:

Z ;

- 1: Reshape K, J, M, Λ_0 to the corresponding vector representation $\tilde{K}, \tilde{J}, \tilde{M}, \tilde{\Lambda}_0$;
- 2: $t = \tilde{K} - \tilde{J}^T \tilde{K} \tilde{M} + \frac{\tilde{\Lambda}_0}{\rho}$;
- 3: $a = \text{abs}(t)$;
- 4: $s = \text{descendingSort}(a)$;
- 5: **while** $i < n$ **do**
- 6: $b_i = \frac{\sum_{j=1}^i s_j - \frac{1}{\rho}}{k}$;
- 7: **end while**
- 8: $\sigma = \max(\max(c), 0)$;
- 9: $\tilde{Z} = \min(a, \sigma) \odot \text{sign}(t)$;
- 10: $Z = \text{reshape}(\tilde{Z})$;
- 11: **return** W and H .

IV. EXPERIMENTS

In this section, we demonstrate the performance of our proposed SCF and RSCF. We conduct clustering experiments on five facial datasets including Yale, YaleB, UMIST, PIE and ORL. In the comparative experiment on SCF, the compared methods in this part involve NMF and its variants including Concept Factorization, Projective NMF [38], symmetric Orthogonal NMF (ONMF, [39]), Graph Regularized Non-negative Matrix Factorization, kernel NMF(KNMF, [9]), Symmetric NMF(SymNMF, [12]), Locally Consistent Concept Factorization (LCCF, [15]). We also compare with some popular clustering methods including Nonnegative Spectral Cut (NSC, [40]), Left-Stochastic matrix Decomposition (LSD, [41]), Data-Cluster-Data decomposition (DCD, [42]), Nonnegative Matrix Factorization Using Graph Random Walk (NMFR, [43]) and Probabilistic latent semantic indexing (PLSI, [44]) as well as classical Kmeans and a recent clustering method Similarity Learning via Kernel preserving Embedding (SLKE, [13]). And for the proposed RSCF, we evaluate the effectiveness comparing with the robust data reconstruction methods including L_{21} NMF [17], CauchyNMF [18], CIMNMF [19], tanhNMF and tanhNMF⁺ [28]. Since some well-behaved baseline methods involve parameters, we use default settings in the compared methods and need no care for parameter free methods. For our SCF and RSCF, we list the concrete parameter settings in Table 1 and Table 2, respectively.

Algorithm 5 ADMM Based Algorithm for RSCF**Input:**

Given nonnegative matrix $X = [x_1, x_2, \dots, x_n] \in d \times n$, positive penalty parameters ρ, λ and γ , increment step s ;

Output:

A basis matrix W and a indicator matrix H ;

- 1: Initialize $\Lambda_0 = \mathbf{0}_{n \times n}$, $\Lambda_1 = \mathbf{0}_{n \times n}$, $\Lambda_2 = \mathbf{0}_{n \times n}$, $J = \mathbf{I}_{n \times n}$, $M = \mathbf{I}_{n \times n}$, $k = 0$, randomly initialize W^k , H^k ;
- 2: **while** not convergent **do**
- 3: $W^{k+1} = OGM_W(W^k, H^k, J^k, M^k, \Lambda_0^k, \Lambda_1^k, \Lambda_2^k, \lambda, \gamma, \rho)$;
- 4: $H^{k+1} = OGM_H(W^{k+1}, H^k, J^k, M^k, \Lambda_0^k, \Lambda_1^k, \Lambda_2^k, \lambda, \gamma, \rho)$;
- 5: Calculate J^{k+1} by Eq. (37);
- 6: Calculate M^{k+1} by Eq. (38);
- 7: $Z^{k+1} = SA(J^{k+1}, K, M^{k+1})$;
- 8: $\Lambda_0^{k+1} = \Lambda_0^k + \rho(Z^{k+1} - K + J^{k+1}KM^{k+1})$;
- 9: $\Lambda_1^{k+1} = \Lambda_1^k + \lambda(W^{k+1}H^{k+1} - J^{k+1})$;
- 10: $\Lambda_2^{k+1} = \Lambda_2^k + \gamma(W^{k+1}H^{k+1} - M^{k+1})$;
- 11: $k = k + 1$;
- 12: $\rho \leftarrow s\rho$;
- 13: $\lambda \leftarrow s\lambda$;
- 14: $\gamma \leftarrow s\gamma$;
- 15: **end while**
- 16: $W = W_{t+1}$; $H = H_{t+1}$;
- 17: **return** W, H .

TABLE 1. Parameter settings of SCF.

Dataset	σ	s
Yale	0.4	1.1
UMIST	4.3	1.1
PIE	0.3	1.1
YaleB	0.1	1.2
ORL	0.1	1.2
MNIST	0.001	1.1
CIFAR-10	0.002	1.9

TABLE 2. Parameter settings of RSCF.

Dataset	ρ	λ	δ	s
Yale	0.1	0.4	0.4	1.1
UMIST	0.53	0.3	0.3	1.01
PIE	0.1	0.1	0.1	1.2
ORL	0.03	0.1	0.1	1.2
YaleB	0.1	0.1	0.1	1.2
MNIST	0.001	0.002	0.001	1.1
CIFAR-10	0.002	0.01	0.01	1.9

Additively, some criteria are set as the following to conduct fair comparison experiments:

- 1) For all the methods, original images have been subjected to the same normalization process and each image is reshaped to a vector. The basis and coefficient matrices of such compared methods are initialized with those of NMF.

- 2) To evaluate clustering performance of the proposed methods, we conduct the clustering experiments with the cluster number ranging from 2 to k . For a fixed cluster number k , we randomly select all the images from the k categories. We conduct 10 times experiment for each cluster number to eliminate the effects of randomization.
- 3) The average clustering accuracy and normalized mutual information is employed to measure the clustering performance evaluation metrics by using the predicted cluster identities and the ground truth. All results are in percentage.

A. DATASETS DESCRIPTION

Our experiments are conducted on the five facial datasets. A brief description is listed in Table 3. We can enjoy the important properties of these datasets as follows:

TABLE 3. The description of the datasets.

Datasets	#size	# Classes
Yale	165	15
YaleB	2414	38
UMIST	564	20
PIE	41368	68
ORL	400	40
MNIST	60K	10
CIFAR-10	60K	10

Yale dataset: The Yale face dataset [45] is created by Yale University and contains 15 people. Each person has 11 facial images and it is total of 165 pictures. The Yale dataset is consisted with different expressions (happy, normal, sad, sleepy, surprised, or wink), poses (with or without glasses) and lighting (center-light, left-light, or right-light). All images are cropped to 32×32 size in our experiment.

YaleB dataset: The Extend Yale B dataset [46] has 38 categories with 168×192 images size. It contains 9 different poses and 64 illumination conditions. All the images are captured in various laboratory-controlled lighting conditions and different facial expressions. An image with background illumination was also captured for every subject in a specific pos. With same operation as [47], this dataset now has 38 individuals and around 64 near frontal images under different illuminations per individual and each image is cropped to 32×32 size. We choose the top 20 classes in our experiments.

CMU Pose, Illumination, and Expression (PIE) dataset: PIE [48] is a database including 41,368 images of 68 people. Each person is under 13 different poses, 43 different illumination conditions, and with 4 different expressions. The face images were captured by 13 synchronized cameras and 21 flashes, under varying pose, illumination, and expression. Similar to [46], we setup a sub dataset containing 10 categories. 170 frontal face images for each category are picked from five near frontal pose classes (C05, C07, C09, C27, C29) and cropped to 32×32 size. All of them are under different illuminations, lighting and expressions.

UMIST: The UMIST face database [49] consists of 564 images of 20 people and has 220×220 pixels in 256 shades of grey. Each category covers a range of poses from profile to frontal views. In our experiment, we pick a subset of left profile facial images of top 10 classes and resize to 40×40 size.

ORL: The ORL (Olivetti Research Laboratory) database of Faces includes 400 images in 40 distinct categories with grayscale 60×112 size. The images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (with / without glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement) [50]. In our experiments, we cropped each image to 32×32 size.

MNIST: The MNIST dataset is a classic dataset in the field of machine learning [51]. It consists of 60,000 training samples and 10,000 test samples. The MNIST is a subset of a larger set available from NIST. The digits have been size-normalized and centered in a fixed-size image. Each sample is a 28×28 pixel grayscale handwritten digital picture. Our clustering experiments are conducted in the test set.

CIFAR-10: CIFAR-10 is a color image data set closer to pervasive objects [52]. It is a small data set for identifying universal objects. A total of 10 categories of RGB color pictures: airplane, automobile, bird, cat, deer, dog, frog, horse, ship and truck. The CIFAR-10 dataset consists of 60,000 32×32 color images in 10 classes, with 6,000 images per class. In our experiment, we use the test images and convert them to grayscale pictures.

B. EVALUATION METRICS

After we obtain the clustering results, the Accuracy and Normalized Mutual Information, two frequently-used external measures, are applied to evaluate the clustering performance by comparing the clustering result with ground truth of each samples.

For each image x_i , r_i and g_i are the corresponding clustering result and ground truth, respectively. The clustering accuracy is signed as:

$$ACC = \frac{\sum_{i=1}^n \delta(r_i, \text{map}(g_i))}{n}, \quad (39)$$

where n is the total number of samples. The permutation mapping function $\text{map}(r_i)$ converts a clustering assignment to its equivalent real assignment and the Kuhn-Munkres algorithm [53] can help to find the best mapping. $\delta(a, b)$ is an indicator function that if $a = b$ is 1, otherwise equals 0. Obviously, a higher clustering accuracy corresponds to a better performance.

Normalized Mutual information (NMI) is used to measure degree of the coincidence between the obtained results and the ground truth clustering. Assume that C and C' are the set of clusters, the Mutual information (MI) between them can be

defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \log \frac{p(c_i, c'_j)}{p(c_i)p(c'_j)}, \quad (40)$$

where $p(c_i)$ and $p(c'_j)$ are probabilities that a data point arbitrarily selected from the data set belongs to the cluster c_i and c'_j , respectively. $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected data point belongs to the cluster c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information as follows:

$$NMI(C, C') = \frac{MI(C, C')}{\max(H(C), H(C'))}, \quad (41)$$

where $H(C)$ and $H(C')$ are the entropies of C and C' , respectively. NMI ranges from 0 to 1. Similarly, the larger NMI, the better clustering performance. If the two sets of clusters are identical, $NMI = 1$; when the two sets of clusters are independent, $NMI = 0$.

C. RESULT AND ANALYSIS

To show the statistically significant clustering results, we conduct experiments ten times on different categories of seven datasets to eliminate the influence of random initialization. We let the average clustering results be the ultimate clustering results. Then clustering ACC and NMI are applied to evaluate clustering performance.

The average clustering ACC and NMI of all compared methods on seven datasets are listed in Table 4 ~ Table 8. It is worth pointing out several observations:

- 1) From Table 4 ~ Table 10, the performance of kernel-based similarity methods is better than the classical clustering methods or graph-based approaches. It can be attributed that the kernel-based similarity retains the global relations of datasets. Our SCF is able to preserve reconstruction relationship measured by kernel-based similarity.
- 2) The NMF variants based on similarity reconstruction is always superior to the standard NMF. This illustrates constraints on data reconstruction or similarity can effectively improve the performance of NMF. In our model, we consider to reconstruct the similarity of raw data meanwhile obtain the data reconstruction and our SCF is surpass to other compared methods in most cases.
- 3) In some datasets, SCF is slightly higher performance than a few methods such as PNMF. However, as the results in Table 4 ~ Table 10, it is not difficult to find that with the comparable performance, our SCF has a lower standard deviation. A lower standard deviation means more stable performance. This also illustrates the efficacy and stability of our algorithm from another aspect.
- 4) All the compared methods achieve the lower results on CIFAR-10. This results from two reasons. First, CIFAR-10 is the color image datasets with RGB

TABLE 4. Average clustering ACC and NMI over Yale dataset.

	<i>k</i>	Kmeans	NMF	CF	SymNMF	DCD	LSD	NMFR	NSC	PLSI	GNMF	LCCF	KernelNMF	SLKE	SCF
ACC	2	66.59±7.41	69.09±16.63	71.64±16.74	76.77±15.81	70.45±10.98	66.82±15.46	68.18±11.54	72.27±12.02	70.45±12.36	72.86±15.78	71.27±15.45	82.27±9.95	65.00±19.29	88.05±7.71
	3	64.06±7.81	66.36±7.76	66.97±10.66	75.67±9.19	58.79±9.39	57.88±12.49	62.73±11.61	60.30±10.73	59.09±8.95	68.70±8.76	66.58±8.76	75.82±9.88	66.67±14.14	85.42±4.67
	4	54.64±9.95	60.91±9.67	57.59±9.34	65.75±14.87	55.45±7.96	50.68±16.91	53.41±8.52	55.02±8.82	55.23±7.19	60.64±8.81	60.27±8.39	66.20±8.34	55.16±14.01	77.45±10.58
	5	49.55±5.39	55.22±7.68	54.75±6.67	58.35±7.71	53.45±8.32	50.91±8.99	52.91±9.17	50.18±9.22	52.91±7.54	55.36±5.34	56.38±7.29	59.11±7.28	48.55±13.21	69.71±8.23
	6	48.94±3.07	51.56±3.45	49.86±4.24	55.97±7.20	47.58±6.55	45.61±9.93	47.88±6.55	48.94±6.23	47.12±6.87	50.23±4.31	52.30±4.74	54.71±5.14	50.74±10.03	64.53±7.90
	7	47.52±4.63	48.68±4.81	47.61±3.50	53.25±3.21	46.62±6.55	47.01±5.22	48.05±6.00	47.40±6.03	45.97±7.14	48.66±2.90	49.21±3.31	52.99±2.97	46.03±10.92	60.73±5.16
	8	48.38±4.58	49.59±5.04	48.95±3.85	54.06±5.21	48.98±4.52	48.84±5.09	48.98±4.02	49.09±4.99	50.11±5.83	50.34±5.42	50.22±5.88	54.31±3.82	50.15±4.50	60.02±4.44
	9	42.91±2.75	45.64±2.87	45.10±3.31	50.83±4.80	43.64±4.25	43.84±8.92	43.64±5.10	44.55±2.91	43.13±9.90	44.68±3.43	44.07±2.72	49.52±1.26	44.36±8.32	55.55±3.26
	10	44.04±2.11	44.99±2.84	44.35±1.81	48.95±4.02	42.73±6.53	41.88±9.24	43.69±4.40	44.82±5.81	41.64±4.61	44.37±3.59	44.97±3.10	49.42±2.63	46.43±4.92	57.45±3.02
	11	44.88±2.80	44.23±1.86	44.27±2.39	50.91±2.50	43.97±2.52	45.70±3.86	44.88±3.17	47.19±2.45	44.21±2.07	44.75±1.66	44.34±2.42	49.17±8.42	45.40±5.00	57.45±3.27
	12	44.17±3.25	42.55±3.16	42.23±3.36	47.51±3.28	42.50±4.13	44.39±5.41	42.12±9.99	43.79±3.23	42.12±4.00	42.26±1.89	41.98±3.15	49.02±2.34	46.55±4.57	55.65±3.64
	13	43.47±0.95	42.57±1.10	43.13±1.66	48.32±2.18	42.80±2.47	46.64±4.44	44.76±2.40	46.15±9.33	43.22±1.05	42.50±1.60	42.41±1.37	49.05±2.10	48.76±4.10	55.50±1.74
	14	42.70±2.19	41.10±1.53	41.78±0.99	46.26±2.23	41.69±2.88	42.86±9.03	44.09±2.78	45.13±2.73	41.43±2.82	41.27±1.58	40.40±2.51	47.32±4.03	45.73±3.02	53.79±2.80
	15	41.84±1.64	39.66±1.80	40.41±1.18	45.53±0.65	42.14±2.51	43.10±2.14	43.41±2.03	43.24±1.67	40.84±0.72	39.13±1.45	38.90±0.88	46.82±2.05	45.42±0.46	52.82±0.36
	Avg.	48.83	50.15	49.90	55.58	48.63	48.36	49.15	49.86	48.39	50.41	50.23	56.17	50.35	63.88
NMI	2	19.99±15.13	23.47±25.88	27.34±27.70	34.17±31.83	16.67±12.77	18.40±16.46	14.84±12.26	19.92±15.02	17.77±14.75	28.40±26.19	26.43±24.43	42.99±25.60	23.19±34.34	53.77±25.69
	3	40.02±16.33	44.81±13.43	46.13±16.70	57.44±17.34	27.36±15.35	30.93±19.74	32.65±17.86	28.02±16.50	27.89±15.18	42.86±14.75	41.18±15.17	51.85±18.61	44.27±20.34	62.81±10.83
	4	38.89±17.24	46.01±14.93	49.35±22.40	47.44±19.42	37.23±14.06	33.02±19.95	34.01±21.17	34.14±12.59	34.10±12.94	42.30±13.22	41.49±13.14	49.52±14.44	41.10±20.40	60.83±14.67
	5	36.72±7.84	44.98±9.50	40.85±7.91	46.20±9.96	36.78±10.79	35.50±12.43	37.42±12.16	33.36±10.69	34.97±10.09	41.41±7.47	41.85±8.72	47.14±8.22	35.99±15.59	53.95±9.57
	6	40.54±5.59	42.89±5.72	39.69±5.46	46.98±8.86	38.20±8.12	36.36±11.73	38.79±6.81	37.64±8.44	40.66±5.42	42.22±6.04	46.36±8.08	46.39±9.63	43.33±9.03	55.83±9.03
	7	41.25±5.74	43.41±5.78	39.90±4.99	47.49±5.08	39.44±1.18	39.35±7.85	39.87±7.71	39.88±4.86	38.84±9.43	41.57±5.08	41.44±9.48	46.41±6.32	42.34±10.61	52.61±4.31
	8	46.43±5.52	45.71±5.51	44.47±4.66	50.66±6.00	44.20±4.84	44.38±6.05	44.62±4.83	45.50±4.29	45.19±5.18	46.05±6.03	45.87±7.15	51.32±4.26	49.15±8.14	55.88±5.62
	9	41.77±3.69	43.06±3.90	41.22±3.62	48.61±4.71	40.83±5.73	40.84±5.73	40.68±5.06	42.02±4.66	40.29±5.58	42.02±4.06	41.18±3.16	48.05±3.16	45.30±5.80	70.14±3.57
	10	45.71±2.32	44.65±3.03	43.13±2.20	49.32±3.63	41.94±9.91	39.48±9.88	42.32±3.72	43.35±4.56	41.07±4.39	43.45±3.21	44.09±5.30	49.53±5.05	49.23±4.71	55.52±2.84
	11	47.79±3.22	45.89±2.48	44.40±2.97	53.48±3.40	45.04±2.92	46.91±3.16	45.22±3.13	48.82±2.30	45.61±2.87	45.72±2.11	45.16±3.06	51.49±4.64	52.00±4.83	57.32±3.51
	12	48.51±0.32	45.15±3.27	43.82±3.26	51.59±3.62	45.09±4.66	45.64±4.20	44.52±3.08	46.21±2.99	44.80±3.66	44.65±2.34	44.24±3.08	51.39±2.89	52.89±3.27	57.18±1.33
	13	49.90±1.46	46.90±1.20	46.04±1.99	54.38±1.32	46.61±4.36	48.38±3.95	47.74±2.54	50.11±3.39	46.85±2.80	45.89±1.29	45.83±1.51	53.80±1.21	56.64±2.19	58.10±1.50
	14	49.10±2.22	46.42±1.34	45.70±2.77	53.29±1.72	45.65±2.48	45.52±8.85	48.05±2.26	49.59±1.99	45.67±2.54	45.36±1.84	44.89±2.04	52.75±2.43	54.54±2.41	57.64±0.23
	15	49.51±1.34	46.03±1.31	45.48±1.00	53.84±0.32	46.41±2.24	48.94±1.40	49.89±1.32	49.60±0.98	45.37±1.12	44.61±1.25	44.57±0.84	52.80±2.08	55.99±0.26	58.10±0.28
	Avg.	42.58	43.54	41.58	49.15	39.39	39.56	39.91	40.71	39.00	42.50	42.17	49.67	46.36	56.49

TABLE 5. Average clustering ACC and NMI over ORL dataset.

	<i>k</i>	Kmeans	NMF	CF	SymNMF	DCD	LSD	NMFR	NSC	PLSI	GNMF	LCCF	KernelNMF	SLKE	SCF
ACC	2	86.75±12.65	89.30±16.90	87.70±16.01	90.45±14.35	86.50±16.67	87.00±17.03	77.50±21.63	88.00±17.98	86.00±16.30	89.55±13.95	88.95±14.08	91.35±11.99	86.50±14.73	92.85±12.43
	4	68.88±8.63	70.10±13.04	69.48±10.47	80.15±13.85	76.25±16.51	76.25±20.01	69.00±13.55	77.28±16.69	77.75±15.96	69.93±11.67	69.53±11.04	76.30±10.87	67.53±14.72	81.83±15.41
	6	65.42±6.80	70.02±9.61	67.35±9.01	76.92±11.25	75.17±18.42	78.50±18.33	74.50±19.04	74.33±10.63	74.33±10.63	71.47±9.91	71.47±9.91	74.33±10.63	68.07±14.50	78.71±14.03
	8	63.05±7.74	67.93±5.76	62.76±6.19	70.10±9.71	70.00±8.23	72.63±11.75	65.25±13.42	71.00±8.12	68.50±8.91	67.59±7.12	67.84±6.91	70.70±9.62	61.01±9.84	74.29±10.30
	10	59.81±2.49	65.58±4.96	61.17±4.01	70.51±7.34	70.20±6.78	70.00±10.46	67.50±9.43	71.30±7.90	70.50±7.95	65.00±2.32	65.83±3.81	68.59±5.62	65.12±8.78	73.57±5.82
	12	58.95±3.73	66.10±3.85	56.83±3.91	69.02±6.75	69.02±6.75	68.33±6.21	69.67±6.81	65.67±4.17	65.45±6.80	64.27±6.43	67.39±6.17	59.59±9.82	74.78±2.30	74.78±2.30
	14	58.81±4.11	66.49±3.55	58.15±4.45	68.22±6.93	67.29±10.03	70.57±7.60	66.79±10.17	68.64±6.67	65.93±8.66	64.51±3.30	66.22±9.32	66.49±7.59	60.67±7.59	72.51±5.89
	16	56.24±4.22	63.09±2.93	54.91±2.90	63.51±5.08	64.56±5.25	66.81±6.25	63.38±4.49	65.69±7.42	63.50±4.45	62.33±3.71	62.38±4.72	64.48±3.94	59.49±4.03	69.63±3.92
	18	56.62±2.60	63.00±2.99	54.84±2.90	63.54±5.29	64.83±5.79	63.11±10.06	64.33±3.94	64.83±5.74	63.06±5.51	64.08±5.01	63.93±5.12	64.87±4.47	63.38±4.15	70.27±4.49
	20	55.97±2.88	62.45±4.22	51.42±3.43	64.06±5.43	65.10±4.35	66.05±7.37	65.60±5.67	65.85±5.25	63.85±4.42	61.58±3.96	61.36±3.67	61.53±2.99	56.32±4.18	70.89±4.54
	22	54.12±3.38	60.40±2.86	50.53±3.21	60.56±5.41	61.32±6.66	63.55±7.38	62.36±5.32	63.05±6.20	61.41±6.30	60.59±4.84	59.66±5.03	60.65±4.57	56.67±5.23	66.84±5.74
	24	52.48±1.75	60.06±2.40	48.03±1.39	61.88±1.83	62.08±1.39	65.42±6.26	63.08±3.90	62.42±3.92	59.92±4.68	60.18±3.47	57.87±2.72	61.08±2.39	52.81±4.56	67.03±3.86
	26	52.29±1.59	58.83±2.91	48.17±2.06	58.15±2.69	60.62±3.79	63.19±4.23	61.04±3.97	62.73±4.04	60.54±1.40	57.38±2.64	54.10±3.40	58.97±3.83	58.17±3.16	66.09±2.43
	28	54.22±1.98	59.03±2.54	47.93±1.73	60.87±2.81	63.04±4.98	64.00±3.65	63.36±3.77	63.25±3.02	61.89±2.95	59.53±3.78	57.08±3.95	61.82±3.82	58.42±4.51	67.73±3.04
	30	53.25±3.24	58.84±2.05	46.53±1.86	57.63±3.24	61.20±3.34	64.17±3.03	60.63±4.22	61.60±2.97	60.90±4.06	56.60±2.14	53.96±3.89	58.64±3.93	50.69±4.60	65.75±3.00
	32	52.53±0.91	57.52±2.07	45.64±1.16	58.26±2.56	59.94±2.06	64.28±2.70	61.41±2.58	62.41±2.56	58.97±2.81	56.01±4.25	52.75±3.35	59.30±3.14	51.98±4.06	66.61±1.44
NMI	2	51.92±1.37	57.67±1.57	46.10±1.21	56.75±2.01	60.97±2.47	62.32±7.43	62.88±3.01	62.50±2.85	61.85±1.60	56.36±1.94	51.76±3.20	59.21±2.79	52.46±4.59	66.16±1.29
	34	51.07±1.27	56.18±1.38	43.36±1.00	55.85±2.16	61.53±2.39	64.91±1.67	60.53±1.75	62.47±1.88	60.69±1.68	55.51±1.34	49.38±2.12	57.81±2.22	47.04±2.58	64.95±0.99
	36	51.60±1.31	55.80±1.11	43.64±0.72	55.38±1.48	61.34±1.80	63.79±2.91	61.05±1.93	62.71±1.62	60.21±1.81	55.50±1.64	49.21±1.37	59.46±1.38	47.22±1.51	65.16±1.08
	40	50.47±0.87	55.77±0.69	42.71±0.84	54.85±0.68	61.45±1.51	63.35±0.75	60.16±1.11	62.27±2.17	59.22±1.78	54.42±1.10	46.33±0.83	58.94±0.95	43.32±0.35	64.58±0.64
	Avg.	57.72	63.21	54.36	64.92	66.16	68.06	64.62	67.11	65.21	62.64	60.69	65.11	58.12	71.03
	2	64.27±33.56	70.49±41.26	67.37±38.80	70.47±40.27	62.03±42.43	63.53±42.71	46.60±46.67	68.81±42.76	59.62±40.68	70.62±34.64	69.11±35.60	75.81±31.33	64.24±38.13	76.92±33.00
	4	60.74±13.68	58.91±16.03	68.64±13.96	69.95±18.16	67.30±19.50	69.67±21.40	58.49±13.14	68.29±19.40	69.52±17.85	68.58±15.45	58.58±17.51	60.40±14.86	63.35±16.57	76.36±17.82
	6	65.79±9.69	72.26±5.95	67.43±10.35	73.78±15.88	76.77±17.11	77.95±19.16	67.99±14.52	73.66±17.26	73.51±16.62	70.47±10.82	71.55±10.66	74.48±10.06	72.29±13.68	78.01±14.03
	8	67.37±8.76	71.81±5.50	67.25±6.37	77.09±9.61	72.52±7.48	73.45±9.65	67.93±10.19	73.09±7.95	71.33±8.07	70.75±6.68	71.24±6.52	73.54±9.94	69.25±8.70	76.69±10.26
	10	67.54±4.55	72.78±4.33	67.62±3.73	73.49±5.17	73.84±5.54	72.77±6.93	71.72±7.46	74.62±6.42	73.70±6.25	71.33±5.12	72.20±3.94	73.65±5.11	74.27±6.00	77.91±4.95
	12	68.05±9.49	73.53±6.67	70.53±6.67	74.10±5.67	74.53±5.67	74.53±5.67	72.71±4.98	74.58±3.66	72.85±3.66	71.85±3.66	71.85±3.66	69.83±5.71	70.93±5.28	77.91±4.95
	14	68.90±4.35	75.65±3.08	67.62±3.18	75.12±4.87	74.94±6.58	76.31±7.97	73.61±7.97	75.62±6.84	73.96±6.67	73.23±2.91	74.00±7.31	71.65±6.17	71.89±6.00	77.84±4.90
	16	68.55±4.15	73.74±2.91	65.74±2.44	72.74±3.71	73.86±4.35	74.82±4.89	72.91±4.92	74.92±4.90	73.16±2.57	72.96±3.15	73.04±3.61	74.44±3.82	73.03±3.79	77.58±2.84
	18	70.14±2.68	75.03±2.51	66.91±2.75	75.13±3.81	75.14±3.84	73.40±7.60	74.23±3.13	74.58±3.63	73.78±3.40	74.06±3.08	74.77±4.03	75.50±4.17	74.78±3.87	79.35±3.58
	20	68.58±2.77	75.09±3.38	64.90±3.03	74.70±4.34	75.36±4.05	75.69±5.22	75.63±4.07	75.88±4.05	74.72±3.77	73.54±3.06	73.15±2.95	73.58±2.96	70.97±3.64	78.86±3.56
	22	69.58±3.37	75.09±3.49	67.50±2.92	75.09±3.49	75.09±3.49	75.09±3.49	74.42±4.28	74.42±4.28	74.42±4.28	74.42±4.28	74.42±4.28	74.42±4.28	74.42±4.28	79.04±2.27
24	68.95±4.18	74.24±1.81	63.53±1.40	74.22±1.27	74.63±2.37	76.60±1.68	74.52±2.39	74.52±2.39	74.00±1.47	73.85±1.55	73.99±2.19	75.03±2.47	69.21±3.45	78.76±2.37	
26	69.14±1.39	73.63±1.98	64.01±1.54	72.07±1.76	74.89±2.21	76.34±2.89	74.29±2.26	75.45±2.41	74.00±2.49	72.15±1.97	69.71±2.55	73.64±3.62	72.78±2.19	78.38±1.57	
28	71.30±1.48	74.87±1.54	64.56±1.78	75.14±2.11	77.31±3.16	77.31±3.16	76.39±3.27	76.39±3.27	74.12±2.93	72.96±3.38	72.96±3.38	76.65±2.42	71.29±2.55	80.09±1.77	
30	70.90±2.21	74.46±1.78	64.01±1.52	72.98±2.21	75.73±2.18	76.75±2.90	74.99±2.16	76.12±1.59	74.89±2.35	73.05±1.57	70.44±3.91	74.53±3.35	68.36±3.50	79.76±1.03	
32	70.90±1.00	74.12±1.54	63.88±0.94	73.63±2.21	77.12±1.41	77.12±1.41	74.72±1.13	74.72±1.13	74.34±1.51	73.11±1.77	70.17±2.01	70.85±2.59	69.95±4.09	79.90±2.83	
34	71.19±1.04	64.48±0.96	64.48±0.96	73.21±1.28	75.41±1.51	75.41±1.51	76.91±1.86	76.91±1.86	76.91±1.86	76.91±1.86	76.91±1.86	76.91±1.86	76.91±1.86	80.12±0.27	
36	71.07±0.89	74.22±1.04	62.80±0.95	72.85±1.40	76.73±1.74	77.69±1.39	75.96±0.97	76.80±1.03	75.94±1.42	73.06±1.09	68.27±1.16	75.50±1.77	69.27±2.20	79.43±0.70	
38	71.60±0.78	74.13±0.73	63.59±0.54	72.66±0.92	77.84±0.89	78.12±1.45	76.72±1.05	76.74±1.13	76.43±0.99	73.54±0.91	68.58±1.72	72.25±0.88	67.64±1.77	80.04±0.74	
40	71.68±0.46	74.21±0.46	63.23±0.78	72.72±0.38	77.48±0.62	78.22±0.50	76.54±0.35	77.32±0.77	75.99±0.69	72.99±0.34	66.65±0.56	77.15±0.60	65.31±0.40	79.76±0.27	
Avg.	68.88	73.14	64.92	73.49	74.32	74.98	71.90	74.88	73.49	72.18	70.56	74.69	69.92	78.69	

TABLE 7. Average clustering ACC and NMI over PIE dataset.

	<i>k</i>	Kmeans	NMF	CF	SymNMF	DCD	LSD	NMFR	NSC	PLSI	GNMF	LCCF	KernelNMF	SLKE	SCF
ACC	2	61.55±11.17	53.25±3.05	51.68±1.66	51.70±1.34	56.00±7.56	56.50±12.59	56.75±7.46	55.75±8.34	56.25±8.27	50.98±2.51	50.83±1.64	72.90±20.11	53.75±1.32	75.75±9.98
	3	46.08±1.67	45.68±5.69	46.78±2.36	49.30±2.53	53.33±1.36	50.00±6.14	45.33±8.56	53.17±5.54	52.17±2.09	48.07±1.80	49.30±2.76	52.22±3.43	38.83±1.93	53.34±5.26
	4	41.00±6.74	40.16±5.99	35.65±3.89	41.28±4.22	36.75±3.40	37.25±2.62	36.63±5.43	34.75±2.87	36.38±3.41	38.58±3.87	37.84±3.89	44.56±6.87	36.00±6.42	43.23±5.00
	5	34.09±7.64	33.57±4.03	30.99±1.89	35.42±8.18	34.20±8.05	32.10±7.06	27.90±4.31	32.00±5.16	33.70±7.78	34.17±5.05	34.41±6.75	39.46±9.08	26.32±1.73	42.69±8.63
	6	28.49±3.44	30.38±2.04	29.09±2.93	28.20±3.12	32.00±3.58	32.92±1.97	30.00±2.99	32.00±3.00	30.67±4.35	29.93±3.69	29.44±3.17	31.03±4.11	26.78±3.74	32.44±4.12
	7	29.16±2.74	28.17±2.32	27.46±2.56	28.87±3.38	29.86±6.37	25.86±1.38	28.07±7.83	31.43±8.62	28.64±4.77	28.83±4.41	29.14±4.39	32.08±4.13	27.49±1.12	34.18±5.73
	8	29.15±1.78	30.33±1.81	27.35±1.48	27.86±1.66	28.38±3.62	27.38±2.70	23.94±5.22	27.44±3.74	26.13±2.57	29.61±1.47	28.93±1.80	31.19±2.63	25.78±3.93	33.88±3.81
	9	26.16±1.76	28.56±2.21	25.09±2.48	27.98±3.32	28.11±2.49	26.17±4.77	27.00±5.28	28.17±3.77	27.72±3.11	29.44±2.94	29.31±2.81	29.43±3.98	22.59±2.09	32.58±2.47
	10	25.53±4.98	26.52±2.27	25.04±1.99	25.92±3.86	25.58±2.33	23.32±4.03	24.71±4.57	25.58±2.13	24.85±1.58	26.60±4.77	27.36±4.59	28.64±3.59	22.47±0.70	31.55±4.94
	Avg.	35.69	35.18	33.57	35.17	34.61	33.37	35.59	35.17	35.13	35.16	39.97	31.11	31.11	42.18
NMI	2	15.48±16.52	2.67±2.96	0.68±0.80	1.57±3.28	2.71±5.75	6.86±19.13	3.08±4.68	2.97±7.14	3.14±7.09	0.85±2.56	0.53±1.30	34.01±31.05	3.87±1.39	33.69±23.70
	3	10.59±13.31	11.75±5.97	12.74±2.61	16.06±3.46	18.88±3.96	17.62±6.95	10.63±7.97	18.96±7.31	19.23±3.76	14.70±1.84	16.42±4.58	18.99±6.47	5.96±2.72	19.10±10.93
	4	15.80±8.07	14.94±5.40	7.22±2.80	13.78±5.40	9.56±4.38	10.74±2.05	8.60±3.72	7.95±4.13	8.41±4.57	9.94±3.67	9.37±3.77	19.38±9.92	11.83±6.25	21.98±8.07
	5	16.20±11.08	16.71±6.32	12.19±3.16	16.62±0.48	15.20±11.43	13.18±10.10	8.79±4.36	13.07±7.75	13.47±11.15	15.74±6.10	15.47±7.71	22.84±13.07	6.35±2.48	27.04±11.44
	6	12.08±4.12	16.93±2.93	13.43±2.84	10.52±3.41	15.47±4.93	17.50±2.28	15.10±1.09	17.52±4.81	13.60±4.78	13.76±4.22	13.29±3.56	14.58±4.10	13.60±5.47	20.41±4.69
	7	18.18±5.43	17.43±4.59	15.07±3.82	16.54±4.86	18.15±10.46	13.18±4.28	15.82±10.01	19.08±10.80	18.12±10.05	16.84±6.19	17.26±6.33	23.11±6.16	14.67±2.78	26.33±7.52
	8	21.93±4.10	23.76±3.89	17.93±2.21	19.14±3.33	18.21±5.12	19.32±3.77	13.39±6.51	18.73±4.83	16.25±3.74	20.81±2.16	19.96±2.78	24.79±2.98	16.71±6.28	29.90±5.12
	9	20.10±2.60	22.46±2.71	21.05±2.72	21.37±3.56	22.64±2.81	19.84±6.81	22.02±5.90	23.46±5.06	22.98±4.23	23.33±4.35	23.62±4.24	24.45±4.50	14.30±3.26	31.33±2.96
	10	21.77±6.92	22.73±3.02	20.23±3.30	21.99±4.84	22.31±3.37	19.98±4.99	21.91±2.94	23.01±2.25	21.76±3.12	22.09±5.01	22.37±5.28	25.46±6.51	17.08±3.62	31.18±6.29
	Avg.	16.90	16.60	13.40	15.29	15.90	15.36	16.26	15.88	15.22	15.34	15.37	23.09	11.60	26.77

TABLE 8. Average clustering ACC and NMI over YaleB dataset.

	<i>k</i>	Kmeans	NMF	CF	SymNMF	DCD	LSD	NMFR	NSC	PLSI	GNMF	LCCF	KernelNMF	SLKE	SCF	
ACC	2	51.54±0.90	51.76±1.02	51.64±1.13	51.87±1.95	67.22±2.36	65.50±6.12	60.19±8.68	55.51±6.83	63.51±7.11	76.12±14.71	60.46±10.44	51.72±1.29	54.50±3.55	67.34±6.13	
	3	36.53±1.63	36.60±1.71	36.63±2.51	37.19±4.04	43.14±5.48	47.28±8.26	43.69±9.67	39.64±5.63	44.44±5.67	52.95±6.92	43.46±8.79	36.68±1.44	39.01±3.55	54.19±3.20	
	4	31.38±2.81	32.68±2.87	35.39±2.12	32.41±3.32	40.91±5.31	44.26±4.87	44.58±9.62	45.81±6.95	40.09±5.43	50.90±7.16	43.03±8.17	31.11±2.95	35.50±5.13	51.83±5.17	
	5	26.65±2.22	30.60±2.88	29.47±2.79	28.44±1.11	36.28±3.28	37.53±3.89	38.33±10.78	41.78±4.29	35.56±3.13	43.01±8.84	32.26±8.98	28.74±3.00	26.72±1.49	46.81±4.18	
	6	25.45±2.09	29.69±2.10	29.02±5.56	28.81±2.92	38.57±6.55	40.95±8.13	33.90±6.45	39.44±4.42	38.73±7.00	39.65±6.67	30.99±5.38	28.04±2.47	27.53±3.39	43.34±2.58	
	7	22.45±1.24	29.86±1.94	26.58±1.85	24.51±2.10	34.10±7.30	38.00±6.23	35.92±5.17	39.10±6.39	34.70±4.11	33.35±4.10	24.86±2.96	23.76±1.34	21.51±1.19	40.78±2.30	
	8	21.25±1.38	27.41±1.74	26.13±2.37	23.66±1.50	33.52±4.93	35.83±5.68	34.01±6.13	37.53±4.43	32.46±4.75	33.59±3.84	24.73±4.79	23.12±1.62	21.92±2.11	40.78±2.30	
	9	19.08±1.74	28.20±1.79	24.78±1.19	20.82±1.79	32.39±3.50	34.84±2.65	32.93±8.02	35.95±5.02	32.77±4.66	29.71±5.23	23.15±4.33	21.18±1.78	19.20±4.13	37.62±1.47	
	10	18.77±1.44	29.00±1.17	25.31±1.33	20.95±1.87	31.90±3.63	34.42±4.00	31.47±4.77	35.60±4.67	32.08±3.63	28.98±6.47	22.17±5.72	20.05±1.64	19.81±0.96	38.11±2.55	
	11	18.01±1.54	27.73±1.48	24.03±1.30	20.20±1.47	30.35±2.60	32.99±4.88	30.33±2.41	34.17±3.97	30.96±2.68	26.83±4.13	20.58±4.06	20.09±1.21	18.59±1.58	36.07±1.62	
	12	17.58±0.78	27.50±1.29	24.46±1.63	18.74±1.24	29.06±2.48	33.97±3.72	27.58±3.62	34.08±4.74	28.62±2.58	26.17±3.75	19.49±2.68	19.09±1.01	18.09±1.68	36.75±1.76	
	13	16.80±1.42	27.96±1.20	24.33±1.49	19.20±1.72	29.52±2.99	34.02±3.65	28.99±6.82	33.95±3.86	29.24±2.53	25.97±3.45	19.08±3.49	19.03±1.40	18.24±1.61	36.20±2.16	
NMI	14	15.89±1.01	28.08±1.29	24.62±0.75	18.22±1.13	29.72±1.31	33.81±2.68	30.02±2.65	34.72±1.88	29.71±2.34	24.05±2.92	17.67±2.29	18.13±0.86	17.29±1.36	35.66±1.61	
	15	15.70±1.08	26.76±1.48	24.28±1.24	17.92±1.31	28.78±2.78	34.42±4.44	26.31±5.05	34.07±5.19	28.26±2.45	23.21±5.18	16.64±2.56	18.17±1.26	16.14±1.91	34.58±2.15	
	16	9.51±0.33	10.56±0.53	13.61±0.58	16.34±0.98	27.31±2.58	30.83±2.90	25.36±4.37	32.69±3.93	28.20±2.07	20.17±3.87	14.82±2.93	17.36±1.27	15.52±1.87	33.78±1.37	
	17	9.24±0.19	10.39±0.55	13.45±1.12	16.38±1.29	26.73±2.07	31.45±3.45	27.43±3.47	31.88±3.13	26.98±1.56	21.64±4.18	15.72±2.93	17.37±1.07	15.96±1.52	33.11±1.72	
	18	9.10±0.18	10.24±0.71	13.41±0.74	15.43±1.71	27.83±1.69	31.12±2.63	27.46±2.33	32.70±2.40	27.15±1.79	19.75±3.50	13.82±2.65	17.13±0.85	15.71±1.47	32.90±1.67	
	19	8.47±0.30	9.64±0.60	13.16±0.67	15.37±0.86	27.22±0.33	30.94±3.35	24.69±4.59	30.69±1.81	26.19±2.33	18.78±2.42	13.75±1.64	16.99±0.69	14.84±1.39	31.74±1.28	
	20	8.34±0.19	9.52±0.33	13.50±0.62	15.10±0.36	27.20±2.16	30.08±2.36	25.45±4.37	33.02±3.33	26.07±2.80	18.59±1.96	13.34±2.04	16.91±0.61	14.92±1.35	33.05±0.97	
	Avg.	20.0922	25.4837	24.9125	23.2392	33.7759	36.9079	33.1226	36.9645	33.4589	32.2851	24.7376	23.4038	22.6840	40.2020	
	NMI	2	0.11±0.10	0.15±0.13	0.15±0.16	0.22±0.43	8.90±2.22	8.10±4.05	6.79±8.32	2.12±3.32	6.74±4.33	36.06±24.07	9.90±13.90	0.16±0.20	0.97±1.38	14.84±7.39
		3	0.67±0.73	0.88±1.57	1.18±2.69	1.89±3.35	8.22±6.05	14.01±10.96	8.43±8.74	4.72±6.09	10.95±6.30	19.19±9.31	9.04±9.55	0.72±0.65	2.80±4.15	20.05±2.91
		4	3.73±3.63	4.81±4.16	10.11±4.45	5.44±4.38	19.58±3.42	21.96±4.85	22.53±6.54	24.09±6.66	18.30±4.70	31.16±6.29	3.42±3.34	6.83±7.27	26.04±5.66	26.04±5.66
		5	4.69±2.36	10.85±5.23	8.18±4.28	8.80±1.98	18.97±5.02	22.82±4.72	17.24±10.66	29.12±2.02	17.95±5.37	29.75±8.97	11.99±11.77	7.66±4.13	3.18±1.43	29.93±2.23
6		8.80±2.97	15.84±2.24	13.72±2.40	14.68±3.79	23.37±7.11	28.68±8.74	18.72±8.03	29.32±4.77	23.91±7.39	28.35±7.86	13.91±5.61	12.34±3.18	9.17±4.00	29.54±2.73	
7		7.16±1.62	20.17±2.69	13.78±2.51	11.52±2.79	21.66±4.54	27.81±7.08	24.77±4.61	32.45±3.19	21.46±3.85	24.14±6.40	11.36±4.83	9.03±2.00	4.84±1.87	28.93±1.58	
8		9.29±1.86	19.56±2.88	15.78±2.36	13.60±1.84	24.94±5.82	28.40±8.52	26.93±4.25	32.59±4.48	23.90±5.44	27.13±5.95	13.34±6.24	10.70±2.12	9.16±1.99	34.50±2.37	
9		7.65±2.48	22.92±1.91	17.04±1.07	10.88±2.95	26.35±3.43	29.48±2.42	28.30±8.66	32.27±5.63	26.99±4.50	24.98±6.80	13.31±6.48	9.50±2.30	7.52±1.87	33.79±2.39	
10		9.23±1.91	24.80±1.66	18.33±1.49	12.43±2.81	26.21±4.80	30.89±5.86	29.23±5.88	33.50±4.23	26.08±5.68	24.15±5.17	13.89±6.87	10.46±2.08	9.85±1.24	33.47±2.51	
11		9.81±2.51	25.23±1.67	19.39±1.70	12.76±2.59	27.47±2.90	30.06±6.88	29.01±2.76	31.31±3.70	26.83±3.03	22.50±4.27	13.62±3.38	12.34±1.64	10.43±1.95	31.73±1.14	
12		10.93±1.75	26.77±2.45	21.07±1.72	12.65±1.83	26.33±3.30	31.75±2.37	29.35±4.79	33.46±4.75	25.47±2.17	23.04±5.61	14.36±3.90	12.56±1.71	11.75±1.80	33.56±1.21	
13		11.00±1.86	28.11±1.62	21.86±2.05	13.96±2.42	27.21±2.94	32.79±1.38	28.89±9.76	32.10±4.59	26.91±3.97	23.95±5.08	14.81±6.18	13.88±1.20	12.57±1.20	34.04±2.15	
14	10.92±1.56	29.78±1.58	23.18±0.84	13.84±1.12	28.11±1.51	34.49±2.31	31.31±2.03	31.24±1.66	27.68±1.77	22.72±3.94	14.27±5.07	13.80±1.26	12.07±1.42	34.51±1.72		
15	11.32±2.21	28.84±1.59	23.67±1.47	14.55±1.77	28.63±3.07	33.71±3.47	30.78±1.82	30.91±1.40	28.04±2.95	22.21±3.74	15.00±2.34	12.44±1.66	13.41±2.56	34.10±2.56		
16	3.01±0.43	4.93±1.06	8.99±1.27	12.79±1.82	14.08±2.67	31.21±4.68	28.60±5.63	30.97±3.60	28.19±2.65	18.87±2.70	10.98±4.69	14.53±1.89	12.31±1.91	34.07±1.85		
17	3.37±0.30	5.58±1.10	10.37±0.88	12.09±1.06	28.47±2.59	34.24±2.79	31.89±4.26	30.52±2.40	27.88±2.44	21.85±2.72	13.95±4.84	16.07±1.30	13.12±1.43	34.52±1.81		
18	3.74±0.30	6.03±1.57	11.62±1.59	12.82±2.29	29.39±2.38	32.02±2.41	33.14±3.25	30.40±1.65	28.24±2.93	19.98±5.25	11.51±4.41	16.59±1.13	13.61±1.88	34.31±1.63		
19	3.33±0.56	5.53±1.01	11.76±1.26	13.49±1.53	27.62±0.93	33.05±4.00	27.27±8.82	30.18±2.81	26.57±3.14	16.58±3.60	12.32±3.44	16.73±1.55	13.34±1.45	33.83±1.70		
20	3.66±0.36	6.01±0.68	13.20±0.96	13.61±0.76	30.48±2.65	34.17±3.50	30.37±1.90	30.02±2.75	29.36±2.35	19.88±2.34	11.30±3.62	17.29±1.06	13.48±1.91	34.81±1.04		
Avg.	6.44	15.10	13.86	11.23	24.22	28.40	25.45	29.76	23.76	23.94	12.95	11.20	11.20	30.90		

TABLE 10. Average clustering ACC and NMI over CIFAR-10 dataset.

	k	Kmeans	NMF	CF	SymNMF	DCD	LSD	NMFR	NSC	PLSI	GNMF	LCCF	KernelNMF	SLKE	SCF
ACC	2	62.35±4.15	62.70±5.51	62.20±4.05	62.02±6.79	58.30±3.65	57.50±6.35	59.50±5.64	55.50±3.75	57.00±4.24	61.19±7.21	64.03±4.75	64.89±5.07	55.70±3.56	65.23±5.05
	3	46.23±2.76	46.45±2.97	46.79±2.66	47.23±2.38	43.80±3.21	41.73±5.96	42.80±1.74	43.40±3.51	43.33±2.69	46.07±5.55	46.59±4.10	47.29±2.63	46.07±7.86	49.23±3.79
	4	38.71±3.00	38.09±2.40	38.89±2.49	40.70±4.02	37.25±3.24	33.35±3.52	33.95±2.62	35.85±2.55	35.90±2.37	39.76±3.08	37.75±2.51	39.93±2.29	37.97±5.64	41.06±3.36
	5	34.28±2.14	35.69±3.25	34.61±2.46	35.52±3.13	30.96±1.66	28.60±4.97	31.84±2.90	31.24±1.31	30.60±1.51	33.40±2.91	32.70±1.63	36.09±3.61	36.84±5.71	38.00±2.51
	6	29.84±1.40	31.42±1.52	30.58±1.51	33.46±0.99	28.00±1.71	27.33±2.24	25.53±2.36	28.27±1.32	27.47±1.39	29.24±1.09	28.93±1.15	31.68±0.94	31.72±5.52	33.27±1.75
	7	27.87±1.34	29.13±1.17	28.16±1.01	28.89±1.79	26.43±1.83	24.60±3.55	25.17±1.51	25.29±1.76	26.23±1.58	26.69±1.35	26.48±1.03	29.19±1.43	31.13±4.44	32.56±1.60
	8	23.92±0.66	25.42±1.35	24.88±1.25	25.25±1.49	22.55±0.97	22.45±1.62	19.90±4.12	22.85±1.90	22.45±1.40	22.50±1.36	23.23±0.82	24.99±0.75	28.55±4.41	29.35±1.34
	9	24.73±1.63	25.55±1.97	25.03±1.74	25.48±2.11	22.73±2.21	22.31±3.10	22.33±1.80	22.13±1.92	22.31±1.85	22.63±1.02	22.70±1.07	25.34±1.85	27.50±4.01	28.43±1.89
	10	24.58±1.25	25.22±1.05	24.59±0.99	25.09±1.52	21.23±1.07	18.86±1.88	19.14±3.31	21.34±1.52	21.28±1.24	21.78±1.05	22.45±0.85	24.95±0.96	25.61±3.46	26.05±1.28
	avg	34.72	35.52	35.08	35.93	32.36	30.75	31.13	31.76	31.84	33.70	33.87	36.04	35.68	38.13
NMI	2	7.04±4.04	5.83±4.35	5.44±4.05	5.69±4.66	2.40±1.70	3.40±2.95	3.68±3.37	1.36±1.39	1.95±1.64	5.45±2.67	8.06±4.17	8.17±5.21	1.36±1.35	8.11±6.10
	3	8.19±2.67	7.73±2.65	8.53±2.66	9.59±2.17	5.87±3.14	5.71±3.36	5.01±2.13	4.37±2.24	5.05±1.86	7.82±2.66	8.85±2.36	8.55±2.67	8.41±10.10	10.92±3.95
	4	9.97±3.36	9.22±2.75	9.72±2.49	10.32±3.67	7.68±1.72	5.04±2.19	4.26±1.26	5.65±1.56	5.12±1.85	10.15±3.00	8.45±2.30	10.46±3.28	7.61±4.42	10.58±3.47
	5	10.56±1.88	11.42±2.37	10.68±2.46	10.84±2.07	6.55±0.89	5.17±2.22	6.27±1.61	5.80±0.95	5.64±0.94	9.98±1.80	9.11±1.42	11.14±2.16	13.05±5.33	13.75±1.54
	6	9.89±1.52	9.73±1.88	10.29±1.51	10.55±1.88	7.26±1.04	6.57±1.03	4.80±1.62	6.61±0.75	6.21±0.95	9.73±1.37	9.14±1.36	10.72±1.29	10.22±5.35	10.96±1.55
	7	10.97±1.76	11.75±2.02	10.57±1.01	11.41±2.11	8.40±1.65	6.78±2.44	6.75±1.22	7.20±1.38	7.26±1.25	10.19±1.29	9.95±1.40	11.74±1.71	10.74±5.98	11.97±1.85
	8	9.27±0.73	10.78±1.07	9.80±1.25	10.78±1.22	7.48±0.96	6.99±1.15	5.04±2.10	7.32±1.58	6.68±1.28	8.49±1.03	8.93±1.01	9.63±0.89	7.38±7.29	10.97±1.12
	9	12.16±2.08	13.37±1.80	12.38±1.74	13.73±2.25	9.30±1.68	8.54±2.59	8.15±1.36	8.93±1.82	9.10±1.28	10.96±1.74	11.42±2.00	12.66±2.15	17.99±4.34	18.16±2.38
	10	14.41±1.47	15.40±1.57	14.15±0.99	15.58±1.10	9.91±1.40	7.85±1.28	8.49±2.56	10.01±1.73	9.81±1.85	12.84±1.37	13.54±1.55	14.76±1.19	18.78±4.27	19.29±1.34
	avg	10.27	10.60	10.17	10.98	7.21	6.23	5.83	6.36	6.31	9.51	9.72	10.87	10.63	12.73

In this subsection, we will discuss the robust of baseline methods including CIMNMF, CauchyNMF, $L_{2,1}$ NMF, tanhNMF, tanhNMF⁺ and our SCF. We conduct our robust experiments also on the three datasets including Yale, UMIST and ORL. We set default parameters of baseline methods according to the description in original papers. As for SCF and RSCF, we set the parameters by Table 1 and Table 2, respectively. In addition, we consider the performance of robust methods after the manual corruption on three facial datasets including Yale, UMIST, PIE. The generated corruption data take the same ways as [28] that randomly select 10% ~ 50% of images and conducts pixel perturbation and key part occlusion simultaneously on these images. We perform corruption on images of the maximum k cluster and select r scale images in Yale to disturb the pixel area of the mouth by using the 8×20 random corrupted weight matrix, while disturb the eyes for UMIST and ORL with a 6×20 random corrupted weight matrix. The other experiment settings are the same. Fig. 3 ~ Fig. 5 plots a few images with/without manual corrupted images.

**FIGURE 3.** The original images(left, after normalization) and eye-corrupted images(right) from PIE dataset.

Fig. 6 ~ Fig. 12 illustrate the performance of compared methods while Table 11 ~ Table 13 shows the average clustering ACC and NMI result versus various proportion of corrupted images. Some facts can be obtained from the shown result in figures and tables as following:

- 1) Our RSCF achieve the satisfactory performance against other robust methods. A higher baseline measured by ACC and NMI is attained. Our model has certain robustness advantages compared with other models.

**FIGURE 4.** The original images(left, after normalization) and mouth-corrupted images(right) from Yale dataset.**FIGURE 5.** The original images(left, after normalization) and eye-corrupted images(right) from UMIST dataset.

- 2) The clustering performance of our RSCF shown in Fig. 6 ~ Fig. 12 also surpasses the other robust methods. It benefits from two aspects. First, our RSCF inherits the advantages of SCF which can reconcile data and similarity reconstruction. Second, the l_∞ norm handles the small errors well which provides robustness. In the figures, we also notice that, in some datasets, such as YaleB, there are slightly surpassing other robust methods. We suppose that after some distortions have been removed, l_∞ norm has a limited effect on residual error.
- 3) Table 11 ~ Table 13 describe the clustering performance on different corrupted images ratio. As r goes up, the performance of robust methods does not fluctuate largely. However, our RSCF is still superior to other robust methods algorithms and this indicates

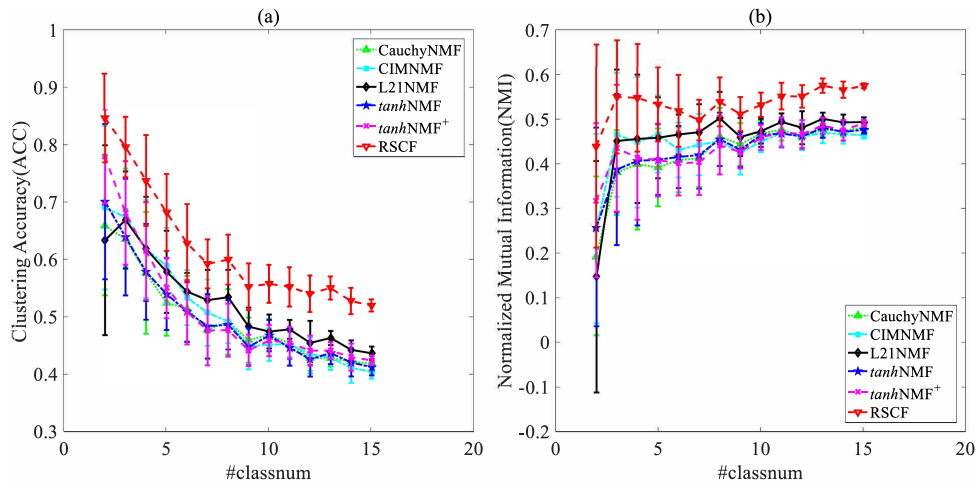


FIGURE 6. Average Clustering ACC (a) and NMI (b) of the compared methods on Yale dataset.

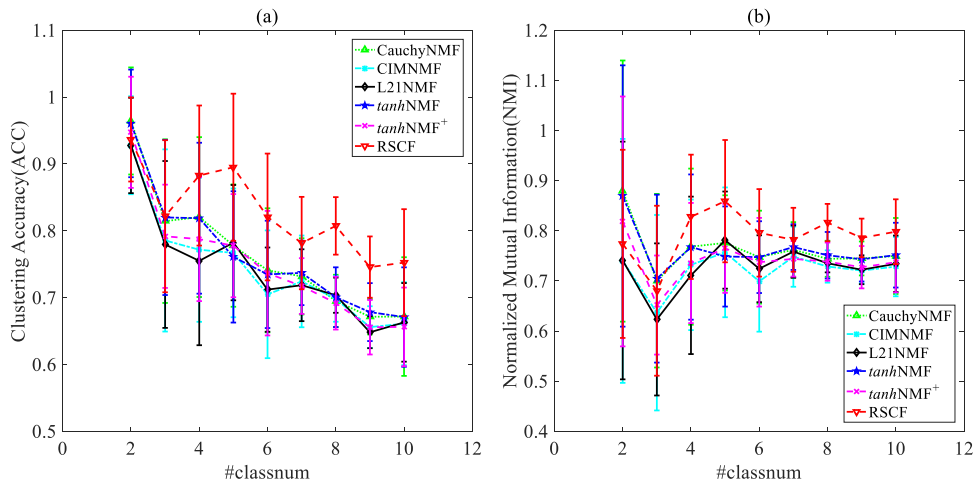


FIGURE 7. Average Clustering ACC (a) and NMI (b) of the compared methods on UMIST dataset.

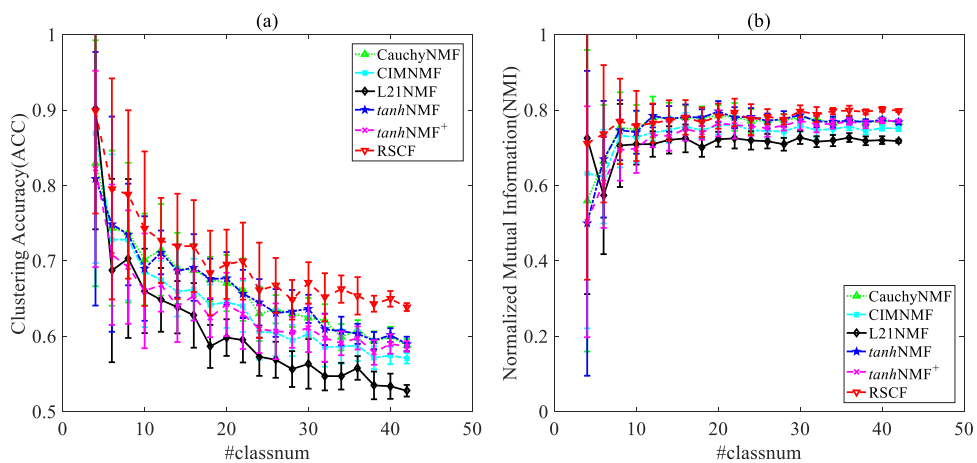


FIGURE 8. Average Clustering ACC (a) and NMI (b) of the compared methods on PIE dataset.

RSCF is less insensitive to the outliers. From these results, we can know that our method is robust to some degree.

4) As Fig. 12 shows, RSCF is slightly superior to the other compared robust NMF methods on CIFAR-10. This could be because the gray images lose some important

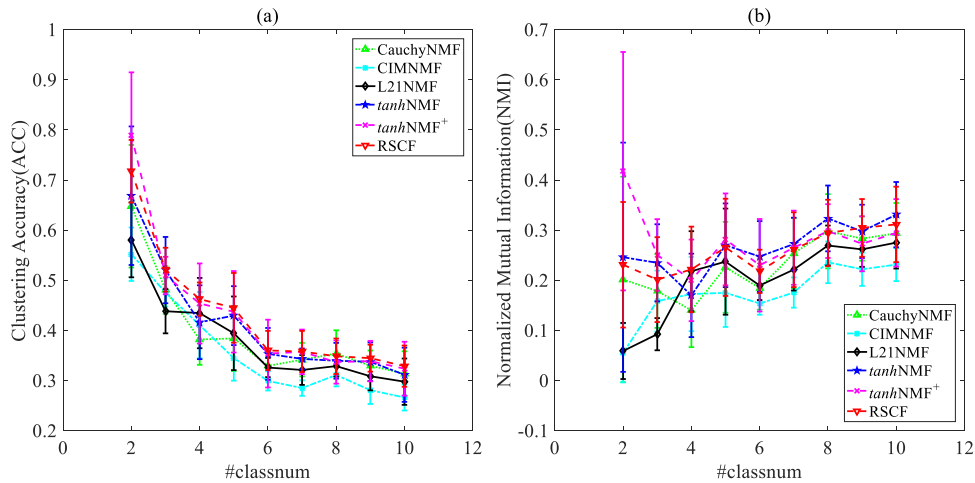


FIGURE 9. Average Clustering ACC (a) and NMI (b) of the compared methods on YaleB dataset.

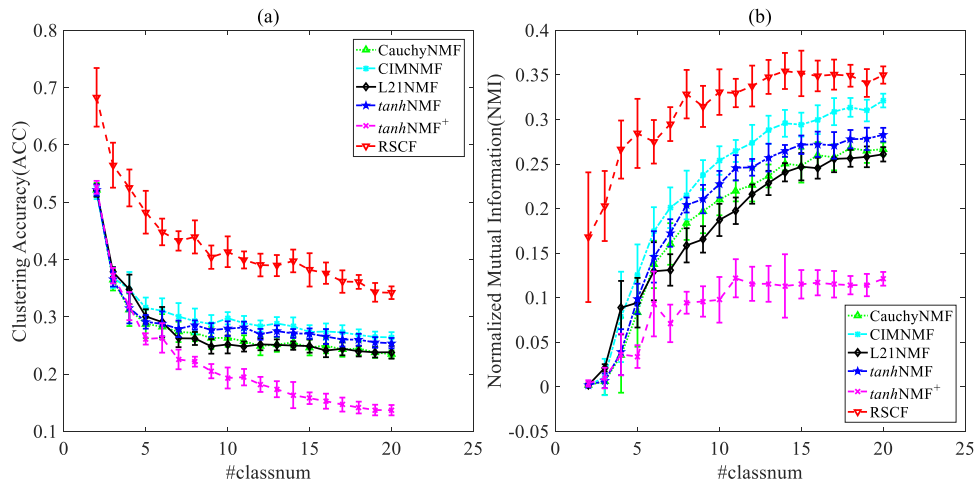


FIGURE 10. Average Clustering ACC (a) and NMI (b) of the compared methods on YaleB dataset.

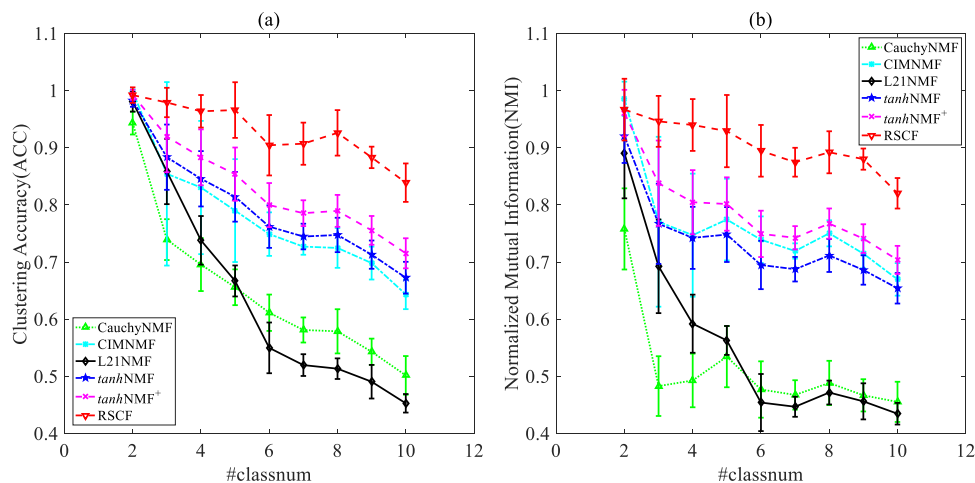


FIGURE 11. Average Clustering ACC (a) and NMI (b) of the compared methods on MNIST dataset.

information. This make robust methods difficult to perform clustering tasks even if they are insensitive to noises.

5) From the shown result in Table 11 ~ Table 13, it is worth noticing that our SCF seems to achieve similar clustering performance to the robust method

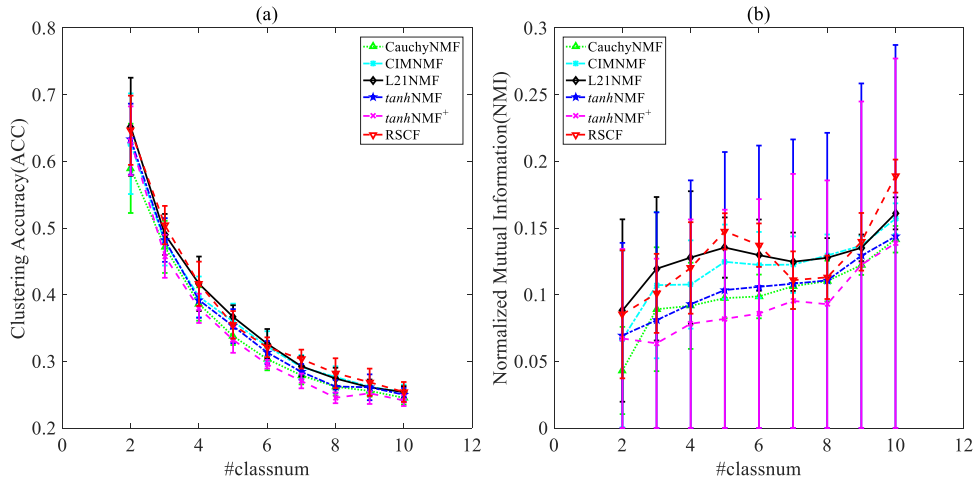


FIGURE 12. Average Clustering ACC (a) and NMI (b) of the compared methods on CIFAR-10 dataset.

TABLE 11. Average clustering ACC and NMI on mouth-corrupted Yale dataset.

	r	CIMNMF	L21NMF	CNMF	TanhNMF	TanhNMF+	SCF	RSCF
ACC	0.1	40.85 $\pm$ 1.46	40.35 $\pm$ 0.91	41.64 $\pm$ 0.87	41.40 $\pm$ 1.32	42.13 $\pm$ 1.20	46.07 $\pm$ 1.85	51.19 $\pm$ 1.61
	0.2	39.57 $\pm$ 1.28	39.24 $\pm$ 1.08	41.49 $\pm$ 1.27	39.93 $\pm$ 1.57	39.90 $\pm$ 1.24	46.54 $\pm$ 1.26	50.32 $\pm$ 1.18
	0.3	39.62 $\pm$ 0.97	39.73 $\pm$ 0.82	42.18 $\pm$ 1.20	39.19 $\pm$ 1.61	38.84 $\pm$ 1.45	45.13 $\pm$ 2.00	47.70 $\pm$ 2.21
	0.4	39.93 $\pm$ 1.16	39.50 $\pm$ 0.86	41.07 $\pm$ 1.51	39.61 $\pm$ 1.21	38.21 $\pm$ 1.12	41.85 $\pm$ 2.65	47.57 $\pm$ 1.99
	0.5	40.32 $\pm$ 0.92	39.04 $\pm$ 0.74	41.73 $\pm$ 1.63	39.45 $\pm$ 1.76	36.84 $\pm$ 1.70	42.28 $\pm$ 1.84	46.75 $\pm$ 2.09
	Avg.	40.06	39.57	41.62	39.92	39.19	43.68	48.71
NMI	0.1	46.94 $\pm$ 1.07	46.54 $\pm$ 0.81	47.55 $\pm$ 1.10	47.60 $\pm$ 1.06	48.39 $\pm$ 1.06	51.46 $\pm$ 1.33	56.32 $\pm$ 1.42
	0.2	45.97 $\pm$ 1.16	45.56 $\pm$ 0.88	47.89 $\pm$ 0.84	46.17 $\pm$ 1.23	46.13 $\pm$ 1.20	49.25 $\pm$ 1.28	54.78 $\pm$ 0.94
	0.3	45.94 $\pm$ 0.69	45.73 $\pm$ 0.76	48.14 $\pm$ 0.99	45.65 $\pm$ 1.15	44.52 $\pm$ 1.65	47.97 $\pm$ 1.65	53.19 $\pm$ 1.27
	0.4	46.00 $\pm$ 1.28	46.02 $\pm$ 0.66	47.73 $\pm$ 1.16	45.86 $\pm$ 1.09	43.86 $\pm$ 1.29	46.92 $\pm$ 1.66	52.19 $\pm$ 1.12
	0.5	46.35 $\pm$ 0.68	45.72 $\pm$ 0.99	48.14 $\pm$ 1.11	45.87 $\pm$ 1.21	42.78 $\pm$ 1.53	46.77 $\pm$ 1.78	51.98 $\pm$ 1.59
	Avg.	46.24	45.92	47.89	46.23	45.14	48.47	53.69

TABLE 12. Average clustering ACC and NMI on eye-corrupted UMIST dataset.

	r	CIMNMF	L21NMF	CNMF	TanhNMF	TanhNMF+	SCF	RSCF
ACC	0.1	66.48 $\pm$ 7.78	65.37 $\pm$ 5.78	63.75 $\pm$ 4.92	67.67 $\pm$ 8.72	65.90 $\pm$ 6.10	65.27 $\pm$ 5.13	67.73 $\pm$ 4.68
	0.2	66.15 $\pm$ 7.42	64.79 $\pm$ 6.31	63.92 $\pm$ 6.27	66.31 $\pm$ 7.64	67.85 $\pm$ 7.09	65.42 $\pm$ 5.64	68.19 $\pm$ 5.68
	0.3	65.01 $\pm$ 6.92	64.91 $\pm$ 5.55	62.96 $\pm$ 5.25	66.95 $\pm$ 7.20	66.30 $\pm$ 8.00	64.73 $\pm$ 5.74	67.45 $\pm$ 5.64
	0.4	64.67 $\pm$ 6.20	65.13 $\pm$ 4.87	62.84 $\pm$ 5.12	65.38 $\pm$ 7.47	64.33 $\pm$ 5.71	63.75 $\pm$ 5.46	67.43 $\pm$ 5.68
	0.5	66.11 $\pm$ 7.38	65.58 $\pm$ 5.86	63.55 $\pm$ 6.05	66.69 $\pm$ 7.20	65.85 $\pm$ 6.22	66.37 $\pm$ 6.49	69.27 $\pm$ 6.03
	Avg.	65.68	65.16	63.40	66.60	66.05	65.16	68.01
NMI	0.1	73.33 $\pm$ 6.49	72.07 $\pm$ 6.14	70.83 $\pm$ 4.75	73.03 $\pm$ 7.32	73.19 $\pm$ 5.98	72.15 $\pm$ 4.57	73.94 $\pm$ 4.24
	0.2	73.45 $\pm$ 6.98	71.31 $\pm$ 5.70	71.17 $\pm$ 5.74	73.74 $\pm$ 6.75	74.70 $\pm$ 6.34	70.65 $\pm$ 5.39	74.04 $\pm$ 5.21
	0.3	73.07 $\pm$ 6.33	71.79 $\pm$ 4.82	70.74 $\pm$ 4.96	73.17 $\pm$ 6.64	72.97 $\pm$ 8.14	71.00 $\pm$ 4.83	73.81 $\pm$ 4.81
	0.4	72.76 $\pm$ 5.66	72.06 $\pm$ 4.82	70.30 $\pm$ 5.31	72.81 $\pm$ 7.00	71.89 $\pm$ 5.45	70.21 $\pm$ 4.98	73.37 $\pm$ 5.07
	0.5	74.09 $\pm$ 6.38	72.25 $\pm$ 5.48	70.77 $\pm$ 5.50	74.95 $\pm$ 6.18	73.01 $\pm$ 5.67	71.77 $\pm$ 6.16	74.58 $\pm$ 5.48
	Avg.	73.34	71.90	70.76	73.54	71.15	70.76	73.95

on the corrupted dataset. We guess that, because of sparse similarity matrix used in our methods, our methods can truncate some larger errors

meanwhile neglect small values. Besides, RSCF significantly improves the clustering performance as compare to SCF. This benefits from the l_∞

TABLE 13. Average clustering ACC and NMI on eye-corrupted PIE dataset.

	r	CIMNMF	L21NMF	CNMF	TanhNMF	TanhNMF+	SCF	RSCF
ACC	0.1	31.06±4.66	26.04±2.94	29.21±3.58	30.46±4.34	30.55±5.51	29.87±4.92	31.71±3.87
	0.2	29.52±4.16	25.76±2.53	28.82±4.15	30.20±3.84	29.91±5.42	29.14±5.36	31.63±4.69
	0.3	29.89±5.14	25.78±2.42	28.77±4.59	28.46±4.92	29.42±5.13	27.99±5.16	30.39±4.87
	0.4	29.94±4.51	25.66±2.01	28.62±4.39	28.50±4.30	29.09±4.88	27.52±5.21	30.19±4.56
	0.5	29.07±5.27	26.04±1.91	28.22±4.31	28.33±3.82	27.94±4.80	27.25±4.51	30.09±4.60
	Avg.	29.89	25.85	28.73	29.19	29.38	28.35	30.80
NMI	0.1	26.91±6.35	23.09±3.26	25.67±3.95	27.54±6.12	29.23±6.77	28.07±6.23	29.67±5.42
	0.2	27.01±5.54	20.94±3.14	25.12±4.00	26.79±5.09	26.22±6.68	26.79±7.68	28.57±6.47
	0.3	26.82±6.82	21.33±2.90	25.02±5.31	26.25±6.70	24.32±5.83	24.44±6.49	27.01±6.59
	0.4	26.10±6.33	21.27±2.81	25.70±4.66	26.30±6.17	23.24±5.97	24.24±7.42	26.65±6.37
	0.5	25.65±6.78	21.32±2.43	25.51±5.11	26.05±4.79	22.80±4.98	24.02±6.84	26.18±6.02
	Avg.	26.49	21.59	25.40	26.58	25.16	25.51	27.62

minimization can suppress the maximum reconstruction error.

V. CONCLUSION

In this paper, we propose a similarity-based concept factorization method called SCF. Our SCF considers similarity reconstruction accompanied by data reconstruction. However, the non-convex of objective function is hard to be optimized. We employ the ADMM optimization framework and OGM algorithm to deal with it. Afterwards, RSCF enjoys the benefit from the insensitivity of l_∞ norm to cope with tiny distortions of similarity reconstruction, which provides a worst-case perspective. Clustering experiments conducted on five popular face datasets with two metrics and 20 methods verify that SCF and RSCF are superior to the compared methods including the variants of NMF and CF.

APPENDIXES

APPENDIX A

PROOF OF COROLLARY 1

Proof: Given two Lagrange multipliers Y_1 and Y_2 , the Eq. (15) is a convex sub-optimization problem for the four primal variables W, H, J, M can be formed as the equivalent problem:

$$\begin{aligned}
 L(W, H, J, M, Y_1, Y_2) &= \|K - J^T K M\|_F^2 \\
 &+ \sum_i^n \left[Y_{1i}^T ((WH)_i - J_i) + \frac{\lambda}{2} \|(WH)_i - J_i\|_2^2 \right] \\
 &+ \sum_i^n \left[Y_{2i}^T ((WH)_i - M_i) + \frac{\gamma}{2} \|(WH)_i - M_i\|_2^2 \right] \\
 &= \|K - J^T K M\|_F^2 + \frac{\lambda}{2} \left\| WH - J + \frac{Y_1}{\lambda} \right\|_F^2 \\
 &+ \frac{\gamma}{2} \left\| WH - M + \frac{Y_2}{\lambda} \right\|_F^2 - \frac{2\|Y_1\|_2^2}{\lambda^2} - \frac{2\|Y_2\|_2^2}{\gamma^2},
 \end{aligned} \quad (42)$$

Obviously, when we optimize one of the variables while the others are fixed, there have the following two sub-optimization problem:

$$\begin{aligned}
 L(W, H) &= \frac{\lambda}{2} \left\| WH - J + \frac{Y_1}{\lambda} \right\|_F^2 + \frac{\gamma}{2} \left\| WH - M + \frac{Y_2}{\gamma} \right\|_F^2
 \end{aligned} \quad (43)$$

$$\begin{aligned}
 L(J, K) &= \|K - J^T K M\|_F^2 + \frac{\lambda}{2} \left\| WH - J + \frac{Y_1}{\lambda} \right\|_F^2 \\
 &+ \frac{\gamma}{2} \left\| WH - M + \frac{Y_2}{\gamma} \right\|_F^2
 \end{aligned} \quad (44)$$

Now we demonstrate the two sub-optimization problem is convex.

Theorem 1: A twice continuously differentiable function f is convex if and only if the second partial derivatives $\nabla^2 f$ is the positive semidefinite (PSD) Hessian matrix.

The proof of **Theorem 1** can be found in [56].

$\nabla_W L(W, H)$ and $\nabla_H L(W, H)$ are the gradient with respect to W and H . They have the following form:

$$\begin{aligned}
 \nabla_W L(W, H) &= \frac{\partial \frac{\lambda}{2} \left\| WH + \frac{Y_1}{\lambda} - J \right\|_F^2}{\partial W} + \frac{\partial \frac{\gamma}{2} \left\| WH + \frac{Y_2}{\gamma} - M \right\|_F^2}{\partial W} \\
 &= \frac{\lambda}{2} \frac{\partial \text{Tr} \left(WH + \left(\frac{Y_1}{\lambda} - J \right) \right)^T \left(WH + \left(\frac{Y_1}{\lambda} - J \right) \right)}{\partial W} \\
 &\quad + \frac{\gamma}{2} \frac{\partial \text{Tr} \left(WH + \left(\frac{Y_2}{\gamma} - M \right) \right)^T \left(WH + \left(\frac{Y_2}{\gamma} - M \right) \right)}{\partial W} \\
 &= \lambda \left(\frac{Y_1}{\lambda} - J \right) H^T + \lambda W H H^T + \gamma \left(\frac{Y_2}{\gamma} - M \right) H^T \\
 &\quad + \gamma W H H^T \\
 &= (Y_1 - \lambda J) H^T + (Y_2 - \gamma M) H^T + (\lambda + \gamma) W H H^T
 \end{aligned} \quad (45)$$

$$\begin{aligned}
& \nabla_H L(W, H) \\
&= \frac{\partial \frac{\lambda}{2} \|WH + \frac{Y_1}{\lambda} - J\|_F^2}{\partial H} + \frac{\partial \frac{\gamma}{2} \|WH + \frac{Y_2}{\gamma} - M\|_F^2}{\partial H} \\
&= W^T (Y_1 - \lambda J) + W^T (Y_2 - \gamma M) + (\lambda + \gamma) W^T WH
\end{aligned} \quad (46)$$

Then we can get the second partial derivatives with respect to W and H :

$$\begin{aligned}
& \nabla_W^2 L(W, H) \\
&= \frac{\partial ((Y_1 - \lambda J) H^T + (Y_2 - \gamma M) H^T + (\lambda + \gamma) WHH^T)}{\partial W} \\
&= (\lambda + \gamma) HH^T
\end{aligned} \quad (47)$$

$$\begin{aligned}
& \nabla_H^2 L(W, H) \\
&= \frac{\partial (W^T (Y_1 - \lambda J) + W^T (Y_2 - \gamma M) + (\lambda + \gamma) W^T WH)}{\partial H} \\
&= (\lambda + \gamma) W^T W
\end{aligned} \quad (48)$$

It is clearly that $\nabla_W^2 L(W, H)$ and $\nabla_H^2 L(W, H)$ are positive semidefinite Hessian matrices. Similarly, $\nabla_J^2 L(J, M) = 2KMM^T K$ and $\nabla_M^2 L(J, M) = 2KJJ^T K$ are also positive semidefinite Hessian matrices. According **Theorem 1**, the two sub-optimization problem is convex. This completes the proof. ■

APPENDIX B PROOF OF COROLLARY 2

Proof: Firstly, we introduced **Theorem 2**.

Theorem 2: Given two metric spaces (X, d_x) and (Y, d_y) , where d_x, d_y denote the metric on the set X, Y , respectively. For any x_1, x_2 in X , a function $f : X \rightarrow Y$ is Lipschitz continuous if there exists a real constant Γ satisfies:

$$d_Y(f(x_1), f(x_2)) \leq \Gamma d_X(x_1, x_2) \quad (49)$$

where $\Gamma > 0$ is referred to a Lipschitz constant for the function f .

Gowers [57] proved the **Theorem 2**. For any two negative matrices $H_1, H_2 \in \mathbb{R}^{k \times n}$, according to the compatibility of the matrix norm, we have:

$$\begin{aligned}
& \|\nabla_H L(H_1) - \nabla_H L(H_2)\|_F^2 \\
&= \|(\lambda + \gamma) W^T W (H_1 - H_2)\|_F^2 \\
&\leq (\lambda + \gamma)^2 \|W^T W\|_F^2 \|H_1 - H_2\|_F^2
\end{aligned} \quad (50)$$

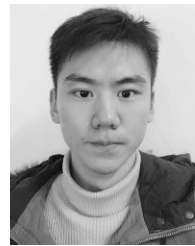
Where the inequality holds because Frobenius norm compatibility. From **Theorem 2**, we can obtain that $\nabla_H L(H)$ is Lipschitz continues and the Lipschitz constant is $(\lambda + \gamma) \|W^T W\|_2$. Similarly, $\nabla_W L(W)$ is also Lipschitz continues and the Lipschitz constant is $(\lambda + \gamma) \|HH^T\|_2$. This completes the proof. ■

REFERENCES

- [1] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, Oct. 1999.

- [2] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.
- [3] L. Lan, N. Guan, X. Zhang, D. Tao, and Z. Luo, "Soft-constrained non-negative matrix factorization via normalization," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 3025–3030.
- [4] N. Guan, D. Tao, L. Lan, Z. Luo, and X. Yang, "Activity recognition in still images with transductive non-negative matrix factorization," in *Proc. Eur. Conf. Comput. Vis.* Springer, 2014, pp. 802–817.
- [5] T. Lerttripinyo, P. Jatukannyaprateep, N. Prompoon, and C. Pattanothai, "Accommodation recommendation system from user reviews based on feature-based weighted non-negative matrix factorization method," in *Proc. 12th Int. Joint Conf. Comput. Sci. Softw. Eng. (JCSSE)*, Jul. 2015, pp. 108–113.
- [6] K. Benzi, V. Kalofolias, X. Bresson, and P. Vandergheynst, "Song recommendation with non-negative matrix factorization and graph total variation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, Mar. 2016, pp. 2439–2443.
- [7] N. Guan, L. Lan, D. Tao, Z. Luo, and X. Yang, "Transductive nonnegative matrix factorization for semi-supervised high-performance speech separation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process. (ICASSP)*, May 2014, pp. 2534–2538.
- [8] C. Ding, X. He, and H. D. Simon, "On the equivalence of nonnegative matrix factorization and spectral clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2005, pp. 606–610.
- [9] D. Zhang, Z. Zhou, and S. Chen, "Non-negative matrix factorization on kernels," in *Proc. 9th Pacific Rim Int. Conf. Artif. Intell.* Springer, 2006, pp. 404–412.
- [10] W. Xu and Y. Gong, "Document clustering by concept factorization," in *Proc. 27th Annu. Int. Conf. Res. Develop. Inf. Retr.*, 2004, pp. 202–209.
- [11] C. Liu, J. Liu, and Z. Jiang, "A multiobjective evolutionary algorithm based on similarity for community detection from signed social networks," *IEEE Trans. Cybern.*, vol. 44, no. 12, pp. 2274–2287, Dec. 2014.
- [12] D. Kuang, C. Ding, and H. Park, "Symmetric nonnegative matrix factorization for graph clustering," in *Proc. SIAM Int. Conf. Data Mining*, Apr. 2012, pp. 106–117.
- [13] Z. Kang, Y. Lu, Y. Su, C. Li, and Z. Xu, "Similarity learning via kernel preserving embedding," *Proc. AAAI Conf. Artif. Intell.*, vol. 33, pp. 4057–4064, Jul. 2019.
- [14] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
- [15] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 902–913, Jun. 2011.
- [16] X. Peng, Z. Yu, Z. Yi, and H. Tang, "Constructing the L2-graph for robust subspace learning and subspace clustering," *IEEE Trans. Cybern.*, vol. 47, no. 4, pp. 1053–1066, Apr. 2017.
- [17] D. Kong, C. Ding, and H. Huang, "Robust nonnegative matrix factorization using L21-norm," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage. (CIKM)*, 2011, pp. 673–682.
- [18] A. Liutkus, D. Fitzgerald, and R. Badeau, "Cauchy nonnegative matrix factorization," in *Proc. IEEE Workshop Appl. Signal Process. Audio Acoust. (WASPAA)*, Oct. 2015, pp. 1–5.
- [19] W. Liu, P. P. Pokharel, and J. C. Principe, "Correntropy: Properties and applications in non-Gaussian signal processing," *IEEE Trans. Signal Process.*, vol. 55, no. 11, pp. 5286–5298, Nov. 2007.
- [20] Y.-X. Wang and Y.-J. Zhang, "Nonnegative matrix factorization: A comprehensive review," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1336–1353, Jun. 2013.
- [21] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Trans. Neural Netw.*, vol. 12, no. 2, pp. 181–201, 2001.
- [22] F. Sha, L. K. Saul, and D. D. Lee, "Multiplicative updates for nonnegative quadratic programming in support vector machines," in *Proc. Adv. Neural Inf. Process. Syst.*, 2003, pp. 1065–1072.
- [23] A. Aghazadeh, A. Lan, A. Shrivastava, and R. Baraniuk, "Rhash: Robust hashing via l_∞ -norm distortion," in *Proc. Int. Joint Conf. Artif. Intell.*, 2017, pp. 1386–1394.
- [24] R. Hartley and F. Schaffalitzky, "L_X minimization in geometric reconstruction problems," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit. (CVPR)*, vol. 1, Dec. 2004, p. 1.
- [25] C. Studer, T. Goldstein, W. Yin, and R. G. Baraniuk, "Democratic representations," 2014, *arXiv:1401.3420*. [Online]. Available: <http://arxiv.org/abs/1401.3420>

- [26] T. Goldstein, M. Li, X. Yuan, E. Esser, and R. Baraniuk, "Adaptive primal-dual hybrid gradient methods for saddle-point problems," 2013, *arXiv:1305.0546*. [Online]. Available: <http://arxiv.org/abs/1305.0546>
- [27] J. Duchi, S. Shalev-Shwartz, Y. Singer, and T. Chandra, "Efficient projections onto the ℓ_1 -ball for learning in high dimensions," in *Proc. 25th Int. Conf. Mach. Learn. (ICML)*, 2008, pp. 272–279.
- [28] X. Shen, X. Zhang, L. Lan, Q. Liao, and Z. Luo, "Another robust NMF: Rethinking the hyperbolic tangent function and locality constraint," *IEEE Access*, vol. 7, pp. 31089–31102, 2019.
- [29] G. Bisson and F. Hussain, "Chi-sim: A new similarity measure for the co-clustering task," in *Proc. 7th Int. Conf. Mach. Learn. Appl.*, 2008, pp. 211–217.
- [30] U. von Luxburg, "A tutorial on spectral clustering," *Statist. Comput.*, vol. 17, no. 4, pp. 395–416, Dec. 2007.
- [31] L. Zelnik-Manor and P. Perona, "Self-tuning spectral clustering," in *Proc. Adv. Neural Inf. Process. Syst.*, 2005, pp. 1601–1608.
- [32] S. Boyd, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Found. Trends Mach. Learn.*, vol. 3, no. 1, pp. 1–122, 2010.
- [33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge, U.K.: Cambridge Univ. Press, 2004.
- [34] X. Zhang, N. Guan, L. Lan, D. Tao, and Z. Luo, "Box-constrained projective nonnegative matrix factorization via augmented lagrangian method," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Jul. 2014, pp. 1900–1906.
- [35] Y. Nesterov, "Smooth minimization of non-smooth functions," *Math. Program.*, vol. 103, no. 1, pp. 127–152, May 2005.
- [36] Y. E. Nesterov, "A method of solving a convex programming problem with convergence rate $O(1/k^2)$," in *Proc. Sov. Math. Doklady*, vol. 27.
- [37] P. Guillaume, J. Schoukens, R. Pintelon, and I. Kollar, "Crest-factor minimization using nonlinear chebyshev approximation methods," *IEEE Trans. Instrum. Meas.*, vol. 40, no. 6, pp. 982–989, Aug. 1991.
- [38] Z. Yang and E. Oja, "Linear and nonlinear projective nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 21, no. 5, pp. 734–749, May 2010.
- [39] C. Ding, T. Li, W. Peng, and H. Park, "Orthogonal nonnegative matrix t-factorizations for clustering," in *Proc. 12th ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 126–135.
- [40] C. Ding, T. Li, and M. I. Jordan, "Nonnegative matrix factorization for combinatorial optimization: Spectral clustering, graph matching, and clique finding," in *Proc. 8th IEEE Int. Conf. Data Mining*, Dec. 2008, pp. 183–192.
- [41] R. Arora, M. R. Gupta, A. Kapila, and M. Fazel, "Clustering by left-stochastic matrix factorization," in *Proc. 28th Int. Conf. Mach. Learn.*, 2011, pp. 761–768.
- [42] Z. Yang and E. Oja, "Clustering by low-rank doubly stochastic matrix decomposition," 2012, *arXiv:1206.4676*. [Online]. Available: <http://arxiv.org/abs/1206.4676>
- [43] Z. Yang, T. Hao, O. Dikmen, X. Chen, and E. Oja, "Clustering by nonnegative matrix factorization using graph random walk," in *Proc. Adv. Neural Inf. Process. Syst.*, 2012, pp. 1079–1087.
- [44] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1999, pp. 50–57.
- [45] K.-C. Lee, J. Ho, and D. J. Kriegman, "Acquiring linear subspaces for face recognition under variable lighting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 5, pp. 684–698, May 2005.
- [46] D. Cai, X. He, and J. Han, "Spectral regression for efficient regularized subspace learning," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, 2007, pp. 1–8.
- [47] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a spatially smooth subspace for face recognition," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–7.
- [48] T. Sim, S. Baker, and M. Bsat, "The cmu pose illumination and expression database of human faces," Carnegie Mellon Univ., Pittsburgh, PA, USA, Tech. Rep. CMU-RI-TR-OI-02, 2001.
- [49] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal laplacianfaces for face recognition," *IEEE Trans. Image Process.*, vol. 15, no. 11, pp. 3608–3614, Nov. 2006.
- [50] D. Cai, X. He, and J. Han, "Using graph model for face analysis," Tech. Rep., 2005.
- [51] L. Deng, "The MNIST database of handwritten digit images for machine learning research [Best of the Web]," *IEEE Signal Process. Mag.*, vol. 29, no. 6, pp. 141–142, Nov. 2012.
- [52] A. Krizhevsky and G. Hinton, "Learning multiple layers of features from tiny images," Tech. Rep., 2009.
- [53] X.-T. Yuan and B.-G. Hu, "Robust feature extraction via information theoretic learning," in *Proc. 26th Annu. Int. Conf. Mach. Learn. (ICML)*, 2009, pp. 1193–1200.
- [54] 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.
- [55] X. Peng, H. Zhu, J. Feng, C. Shen, H. Zhang, and J. T. Zhou, "Deep clustering with sample-assignment invariance prior," *IEEE Trans. Neural Netw. Learn. Syst.*, early access, Dec. 31, 2019, doi: [10.1109/TNNLS.2019.2958324](https://doi.org/10.1109/TNNLS.2019.2958324).
- [56] H. Tuy, T. Hoang, T. Hoang, V.-N. Mathématique, T. Hoang, and V. Mathématique, *Convex Analysis and Global Optimization*. Cham, Switzerland: Springer, 1998.
- [57] W. T. Gowers, "Lipschitz functions on classical spaces," *Eur. J. Combinatorics*, vol. 13, no. 3, pp. 141–151, May 1992.



XINGYU SHEN received the B.S. degree from Lanzhou University, Lanzhou, China. He is currently pursuing the master's degree with the National University of Defense Technology, Changsha, China. His current research interests include computer vision and machine learning.



XIANG ZHANG received the M.S. and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 2010 and 2015, respectively. He is currently a Research Assistant with the State Key Laboratory of High Performance Computing, College of Computer, Institute for Quantum Information, National University of Defense Technology. His current research interests include computer vision and machine learning.



LONG LAN (Member, IEEE) received the Ph.D. degree in computer science from the National University of Defense Technology, in 2017. He was a Visiting Ph.D. Student with the University of Technology Sydney, from 2015 to 2017. He is currently a Lecturer with the College of Computer, National University of Defense Technology. His research interests include multi-object tracking, computer vision, and discrete optimization.



QING LIAO (Member, IEEE) received the Ph.D. degree in computer science and engineering from the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, in 2016, supervised by Prof. Q. Zhang. She is currently an Assistant Professor with the School of Computer Science and Technology, Harbin Institute of Technology (Shenzhen), China. Her research interests include artificial intelligence and bioinformatics.



ZHIGANG LUO (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from the National University of Defense Technology, Changsha, China, in 1981, 1993, and 2000, respectively. He is currently a Professor with the College of Computer, National University of Defense Technology. His current research interests include machine learning, computer vision, and bioinformatics.

...