Robust Unsupervised Flexible Auto-weighted Local-coordinate Concept Factorization for Image Clustering

Author Names

Affiliations and Addresses   
publications18@aaai.org

Abstract

We mainly discuss the high-dimensional data clustering problem by proposing a novel unsupervised representation learning model called Robust Flexible Auto-weighted Local-coordinate Concept Factorization (RFA-LCF). To improve the representation and clustering abilities, RFA-LCF explicitly integrates the robust flexible CF, robust sparse local-coordinate coding, error correction, adaptive reconstruction weight learning and the joint manifold preserving constraint on the recovered clean, basis concepts and new representation into a unified model. Specifically, RFA-LCF uses a L2,1-norm based flexible residue to encode the mismatch between the error corrected data and its reconstruction, and also uses the robust adaptive sparse local-coordinate coding to represent the data using a few nearby anchor points or basis concepts, which can make the factorization more accurate and robust to noise. The error correction step learns an explicit sparse projection to remove noise from data, and then the robust flexible CF performs in the clean data space for enhancing the data representations. RFA-LCF also considers preserving the local manifold structures of the clean data space, basis concept space and the new coordinate space jointly in an adaptive manner by integrating the minimized reconstruction error over the clean data, anchor points and the coordinates jointly. Extensive comparisons show that RFA-LCF can deliver the enhanced clustering results.

Introduction*[[1]](#footnote-2)*

Clustering images via the efficient representation is always a fundamental issue, but real images usually has redundant information and noise that may decrease the representation and clustering abilities directly. In recent decades, lots of effective matrix factorization based algorithms have been proposed for the data representation, among which Vector quantization (VQ) (Gray, 1984), Singular Value Decomposition (Golub et al. 1970), Principal Component Analysis (PCA) (Jolliffe, 1986), Nonnegative Matrix Factorization (NMF) (Lee et al. 1999), and Concept Factorization (CF) (Wei et al. 2004) are several classical algorithms. Among these factorization approaches, NMF and CF clearly differ from others since they imposes the nonnegative constraints on the factorization matrices explicitly.

Due to the additive nonnegative constraints, NMF and its variants, for instance Projective NMF (PNMF) (Yuan et al. 2010), Graph Regularized NMF (GNMF) (Cai et al. 2011), Constrained NMF (CNMF) (Liu et al. 2012), Dual-graph Sparse NMF (DSNMF) (Meng et al. 2018), Graph dual regularization NMF (DNMF) (Shang et al. 2012), and Parameter-less Auto-weighted Multiple Graph regularized NMF (PAMGNMF) (Shu et al. 2017) are widely used for learning the parts-based representation for representing and clustering the faces, documents and texts, and the enhanced result are obtained. But NMF and its variants cannot deal with data in the kernel space directly. To handle this issue, CF was recently proposed to represent each data by a linear combination of cluster centers, and thus can be performed in any representation space, e.g., original space and kernel space. Like NMF, CF can only reveal the global geometry but cannot preserve the local manifold structures. To tackle this drawback, recently several effective locality preserving enhanced variants have been proposed, such as Locally Consistent CF (LCCF) (Cai et al. 2011), Local Coordinate CF (LCF) (Liu et al. 2014), Graph-Regularized Local Coordinate CF (GRLCF) (Ye et al. 2017), Graph-regularized CF with Local Coordinate (LGCF) (Li et al. 2017) and the Dual-graph regularized CF (GCF) (Ye et al. 2014).

Although the enhanced representation results have been produced by aforementioned manifold preserving CF variants, they still suffer from several obvious drawbacks. First, to preserve the locality of representations, LCCF, GRLCF, LGCF and GCF usually search the neighbors of each sample by *k*-neighborhood or *ε*-neighborhood firstly, and then pre-compute the graph weights by a separable step before factorization. But estimating an optimal *k* or *ε* still remains a tricky issue in the real applications (Roweis et al. 2000; Tenenbaum et al. 2000), and using the same *k* or *ε* for each sample is also unreasonable as real application data usually have complex distribution (Belkin et al. 2002). Moreover, the pre-calculated weights and graph Laplacian prior to the factorization process also cannot be ensured as optimal for seeking the low-dimensional representation of original data explicitly. Second, the steps of searching neighbors, defining weights and factorizing data of aforementioned methods are all performed in the original input space, but real data usually has noise, redundant information and unfavorable features that may cause negative effects on the results, such as the inaccurate similarities and representations. As a result, it would be better to weight and represent data in a recovered clean space, which can potentially produce more accurate and compact data representations. Third, although CF and its variants employ a residue term to minimize the reconstruction error between original data and the product of three factors as a hard constraint for discovering the new representation of original data, we still argue that such an operation assumes that the new representation should lie in the nonnegative space and a linear combination of the cluster centers should be able to represent each data point, but these hard constraints may be over-strict in practical applications. For example, some real application data may have a nonlinear manifold structure. In such case, the results by the linear reconstruction may be inaccurate in practice.

In this paper, we therefore propose a novel and robust adaptive locality preserving flexible CF method. The major contributions of this paper are summarized as follows:

(1) Technically, a novel and unsupervised Robust Flexible Auto-weighted Local-coordinate Concept Factorization framework termed RFA-LCF is proposed. RFA-LCF enhances the data representation ability by explicitly improving the robust properties of factorization to noise and errors by jointly recovering clean data, enhancing the similarities by adaptive weight learning, and providing flexible residue for encoding the mismatch between the original data and the product by relaxing the hard constraint so that the samples resided on a nonlinear manifold can also be processed potentially. Specifically, RFA-LCF incorporates the robust flexible concept factorization, robust adaptive sparse local co-ordinate coding, error correction, auto-weighting learn-ing, and joint manifold constraints on the clean data, basis vectors and new coordinates into a unified framework.

(2) For robust flexible concept factorization, RFA-LCF improves the representations in twofold. First, it enhances the robust properties to noise and outliers by involving an error correction process to obtain an explicit projection to remove noise from data, and then conducts factorization in the clean data space for enhancing the representations. Besides, RFA-LCF uses the sparse L2,1-norm to encode the reconstruction loss between the recovered clean data and its reconstruction, since L2,1-norm is robust to noise and outliers and moreover has a potential to minimize the loss (Yang et al. 2011). Second, RFA-LCF considers relaxing aforementioned mismatch to handle the data sampled from a nonlinear manifold, inspired by (Nie et al. 2010). That is, RFA-LCF applies a flexible penalty term on the factorization loss by relaxing the existing assumption that each data point might be represented by a linear combination of the cluster centers, which is clearly a soft constraint.

(3) To guarantee the encoded locality and sparsity to be more accurate, RFA-LCF also integrates the adaptive reconstruction weight learning with the robust flexible CF to discover not only the manifold structures of the given data as LCCF, but also the manifold structures of the basis concept vectors and new representations in an adaptive manner at the same time. Moreover, RFA-LCF also uses the robust adaptive sparse local coordinate coding performed in clean data space to represent the data by using a few most nearby basis concepts, which is different from LCF that performs the local coordinate coding in the original data space. Thus, the new representation by our RFA-LCF will be potentially more accurate, informative and robust to noise.

Related Work

In this section, we briefly review LCCF and LCF that are closely related to our proposed formulation.

**LCCF.** Given a dataset, where  denotes a sample vector,  is the number of samples and *D* is the original dimension, LCCF learns the locality preserving new representa­tion of *X* by adding a geometrically based regularizer. Letand  be two nonnegative matrices whose product is the approximation to data *X*, where the rank *R* is a constant. LCCF first constructs a graph *G* with *N* nodes based on *X*, where each vertex in vertex set corresponds to, and the weight for edge connecting and  is defined as



, (1)



whereis the set including the *k* nearest neighbors of each data *xi*. By representing each basis by a nonnegative linear combination of, i.e.,, where, the objective function of LCCF can be defined as



, (2)



whereis the regularization term, graph Laplacian , *D* is a diagonal matrix whose the entries being .  is a regularized weighting factor.



**LCF.** Different from LCCF, LCF is inspired by the idea of the local coordinate coding. LCF takes the locality constraints as an additional requirement, considers the basis vectors by CF, the anchor points  and the coordinates for each sample over each column of *V* with respect to the anchor points. Then, LCF defines the following constraint to measure the locality and sparsity penalties between the anchor point  and:

, (3)

Thus, LCF minimizes the following objective function:

, (4)

whereis a weighting parameter. That is, LCF tries to represent  using only a few nearby anchor points so that the sparse and local structure can be preserved.

Robust Flexible Auto-weighted Local-coordinate Concept Factorization

The Objective Function

We present the formulation of our RFA-LCF in this section. Given the dataset, RFA-LCF jointly calculates a L2,1-norm based sparse projection  to remove the noise and outliers in data by embedding *X* onto it directly, and then runs the factorization over the recovered clean data to calculate two nonnegative matricesandso that the product of *X*, *W* and , that is, , can approximate the recovered clean data. Clearly, RFA-LCF performs the concept factorization based on the clean data space spanned by using *P* rather than the original input space *X*, which can potentially make the factorization process more accurate and robust. For the robust flexible CF, RFA-LCF set the factorization based on the clean data as, where  is a transform function for factorizing data. Assuming that is the linear regression function, where is a column vector of all ones and is the bias vector, then  can encode the mismatch between  and. To make the residue  more accurate, the sparse L2,1-norm is also imposed on it, i.e., . To encode the neighborhood information and pairwise similarities more accurately, our RFA-LCF encodes the manifold structures jointly over the clean data, basis concept vectors and new coordinates in an adaptive manner by minimizing the joint reconstruction error them explicitly, i.e., , where *Q* is the reconstruction weight matrix. RFA-LCF also involves the robust adaptive neighborhood preserving local coordinate coding to represent data using a few most nearby basis concepts, which can potentially make the factorization result more informative. These discussions can lead to the following objective function for RFA-LCF:

, (3)

where are the nonnegative constraints, is added to avoid the trivial solution, and are trade-off parameters. Since L2,1-norm can force the residue to be sparse in rows and robust to noise (Yang et al. 2011), so minimizing the L2,1-norm based flexible residue has a potential to reduce the reconstruction error. denotes the robust adaptive locality and sparsity constraint term and  is auto-weighted learning term, defined as

. (4)

To highlight the benefits of involving and, next we briefly discuss the sum of them as follows:

, (5)

from which one can find that the neighborhood relationship can also be encoded in an adaptive manner by integrating the reconstruction error  based on the basis concept vectors and new coordinatesinto the local coordinate coding. In addition, RFA-LCF performs coordinate coding in recovered clean data space to represent data. Thus, RFA-LCF involves a robust adaptive neighborhood preserving locality and sparsity constraint penalty between the anchor point  and.

Note that the formulation of our RFA-LCF can be performed alternately between the following three steps.

1) Robust Flexible Auto-weighted Local-coordinate CF.

The projection  and adaptive weight matrix *Q* are fixed in this step, and we can focus on the robust flexible adaptive local-coordinate CF for representing data. With *P* and *Q* fixed, we have the following reduced formulation:

,(6)

where  and  trades off the robust adaptive sparse local-coordinate and adaptive neighborhood preservation. After *W* and *V* are obtained, we update the sparse projection *P*.

2) Error Correction by Robust L2,1-norm Projection.

We focus on seeking the robust projection *P* for removing the noise and outliers in data, with *W*, *V* and *Q* given. The sub-problem involved in this step can be formulated as

. (7)

After the clean data is updated by the new *P*, we can return it for the robust flexible adaptive local-coordinate CF and the following graph re-weighting process.

3) Auto-weighted Reconstruction Graph Learning.

We focus on the auto-weighted reconstruction graph learning to preserve the manifold structures of clean data, basis vectors and new representations explicitly and adaptively. The auto-weighted learning term can be formulated as

, (8)

where the entry *Qi,j* measures the contribution of *xj* to reconstruct each sample *xi*.

Optimization

We show how to optimize the objective function of RFA-LCF. Since there are several variables in RFA-LCF and the involved variables depend on each other, its problem cannot be solved directly. Thus, we follow the common procedures to update the variables alternately. Let  be the objective function, and be a diagonal matrix with diagonal entries being. According to the properties of L2,1-norm (Yang et al. 2011), we can have

. (9)

By taking the derivative of Eq.(3) *w.r.t.* bias *b* and setting the derivative to zero, we can easily obtain

, (10)

whereis a constant. Thus, the flexible residue can be rewritten as

, (11)

whereand. Then we have , , where *S* is a diagonal matrix with entries ,  is the *i*-th row vector of *P*. Suppose each and  over each index *i* and let, we can have the matrix trace based problem for RFA-LCF:

, (12)

where and . Then, the optimization of our RFA-LCF can be described as follows:

1) Fix *others*, update the matrix factors *W, V*.

We first show how to optimize *W* and *V*. Letand be the Lagrange multipliers for nonnegative constraints, andrespectively, and, then the Lagrange functionof Eq.(12) can be constructed as

. (13)

By taking the partial derivatives of with respect to *W* and *V* respectively, and by using the Karush-Kuhn-Tucker conditions, and, and letting, we can easily obtain the following two equations:

,(14)

,(15)

where *A* represents a matrix whose rows are  , and *B* is a matrix whose columns are. Note that the above equations can lead to following multiplicative updating rules for the basis vectors *W* and new representation *V*:

, (16)

. (17)

2) Fix *others*, update *P* for error correction.

By removing the terms that are irrelevant to variable *P* from Eq.(12), we can obtain the following reduced formulation:

.(18)

By taking the derivative of *w.r.t. P*, setting the derivative  to zero, we can update projection *P* as

, (19)

where and *E* is an  matrix of all ones. After *P* is updated, we can use it together with the factors *W* and *V* to compute the adaptive weight matrix *Q*.

3) Fix *others*, update the adaptive weighting matrix *Q*.

By removing the irrelevant terms to *Q* from Eq.(12), we can obtain the following reduced formulation:

, (20)

where. Letbe the Lagrange multiplier for the nonnegative constraint, and, the Lagrange function of Eq.(20) can be constructed as

. (21)

By taking the derivative of with respect to *Q*, and using the KKT condition, we can obtain the following updating rule for *Q*:

. (22)

|  |
| --- |
| **Algorithm 1:** Our Proposed RFA-LCF Framework |
| **Inputs:** Training data matrix, control parameters, and the constant *R* (rank of the factorization).  **Initialization:** Initialize the weight matrix *Q* by the cosine similarity, i.e., ; Initialize variables *W* and *V* to be random matrices; .  ***While not converged do***  1. Update *W* and representation *V* by Eqs.(16) and (17);  2. Update the adaptive weight matrix  by Eq. (22) ;  3. Update the robust projection *P* by Eq. (19) ;  4. Check for convergence: if, stop; else . ***End while***  **Output:** Learnt new representation. |

To present our algorithm completely, we summarize the procedures of RFA-LCF in Algorithm 1, where the diagonal matrices *M* and *S* are initialized to be the identity matrices similarly as (Yang et al. 2011) so that each and  over each index *i* is satisfied.

Convergence Analysis

The problem of our RFA-LCF is solved alternately, so we would like to describe the convergence analysis. We summarize the convergence analysis in Proposition 1. To assist the proof, a lemma in (Nie *et al.* 2010) is first reviewed.

**Lemma 1:** For any pair of nonzero vectors, the following inequality holds:



. (23)

**Proposition 1.** The procedures in Algorithm 1 monotonically decrease the objective function at each iteration.

**Proof.** As seen from the procedures in Algorithm 1, when we fix *S* as *St* at the *t*-th iteration and update *Wt+1*, *Vt+1*, *Pt+1*, *Qt+1,* the following inequality holds:

, (24)

where the auxiliary matrixis defined as



. (25)

Since, the following inequality holds:

. (26)

According to the result in Lemma 1, it is easy to obtain

. (27)

Let and by using the above inequalities, we can obtain the following inequality easily:

, (28)

which explicitly indicates that the objective function value of RFA-LCF is decreased monotonically in the optimizations. Note that the Proposition 1 clearly indicates that the objective function is non-increasing, but it is also vital to show that the new representation *V* also converges, since it is one of the major variables for clustering and other variables also depend on *V*. The convergence condition is simply set as  and=, which measures the difference between two sequential new representations and thus it can ensure the result will not change drastically.



Computational Complexity Analysis

We mainly describe the extra cost of RFA-LCF in comparison to LCF. We also use big O notation to show the complexity (Cormen et al. 2009). According to the updating rules in Eqs.(16-17), we can conclude that RFA-LCF has the same computational time complexity as LCF by using the big O when updating *W* and *V*, i.e., O(*N2R*), where *N* is the number of *X* and *R* is the dimension of new representation *VT*. Besides, we also need to compute *P* and *Q* in each iteration. According to the Eqs.(19)(22), the complexity of updating *P* and *Q* is O(*d3*), where *d* is the dimension. Thus, the overall cost will be O(*tN2R*) when *N* is far large than *d*, when the updates stop after *t* times iterations.

Simulation Results and Analysis

We conduct simulations on several real image databases to examine RFA-LCF for data clustering by representation. The results of our RFA-LCF are compared with those of 12 related nonnegative factorization algorithms, i.e., NMF (Lee et al. 1999), PNMF (Yuan et al. 2010), GNMF (Cai et al. 2011), DNMF (Shang et al. 2012), DSNMF (Meng et al. 2018), PAMGNMF (Shu et al. 2017), CF (Wei et al. 2004), LCCF (Cai et al. 2011), LCF (Liu et al. 2014), LGCF (Li et al. 2017), GRLCF (Ye et al. 2017) and GCF (Ye et al. 2014), which are closely related to our method. Note that there are no parameters in NMF, PNMF and CF, and parameters of GNMF, DNMF, DSNMF, PAMGNMF, LCF, LCCF, GRLCF, LGCF and GCF are carefully chosen for fair comparison. Six public image databases are evaluated, including one object database, i.e., COIL100 (Nayar et al. 1996), three face databases, i.e., UMIST (Wechsler et al. 2012), ORL (Lu et al. 2003) and CMU PIE (Sim et al. 2002), two handwritten datasets, i.e., digit and letter subsets from CASIA-HWDB1.1 (Zhang et al. 2015), where HWDB1.1-D includes 2381 handwritten digits (‘0’-‘9’) of 14×14 pixels and HWDB1.1-L contains 12456 handwritten letters (‘a’-‘z’ and ‘A’-‘Z’) of 16×16 pixels from 52 classes. For the face and object recognition databases, all images are resized into 32×32 pixels (i.e., each image is represented by a 1024-dimensional column vector) for computational efficiency. The information of the used datasets are shown in Table I. We perform all the simulations on a PC with Intel Core i5-4590 CPU @ 3.30 GHz 3.30 GHz 8G.

**Table I.** List of used datasets and dataset information.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| ***Data Type*** | ***Dataset Name*** | ***# Points*** | ***# Dim*** | ***# Class*** |
| **Face**  **images** | ORL | 400 | 1024 | 40 |
| UMIST | 1012 | 1024 | 20 |
| CMU PIE | 11554 | 1024 | 68 |
| **Object**  **images** | ETH80 | 3280 | 1024 | 80 |
| COIL100 | 7200 | 1024 | 100 |
| **Handwritten images** | CASIA-HWDB1.1-D | 2381 | 196 | 10 |
| CASIA-HWDB1.1-L | 12456 | 256 | 52 |

Visualization of Graph Adjacency Matrix

We compare the adaptive weighting matrix Q of RFA-LCF with the binary weights applied in GCF and DNMF, Cosine similarity weights used in LCCF, and the CLR weights (Nie et al. 2016) used in GRLCF. The ORL face database is used as an example. We choose face images of 10 people to construct the adjacency graphs, and the number of nearest neighbors is set to seven (Sugiyama et al. 2007) for each weighting method for the fair comparison. The weight matrices are illustrated in Fig.1. We can find that: 1) The constructed weight matrices by each weighting approach have approximate block-diagonal structures; 2) more noisy or wrong inter-class connections are produced in the binary weights, Cosine weights and the CLR weights, which may potentially result in the inaccurate similarity measures and high clustering error, compared with our adaptive weights.



(a) Cosine Similarity weights (b) Binary weights

(c) CLR weights (d) our adaptive weights

*Fig.1: Visualization comparison of the constructed weights by each weighting approach on the ORL face database.*

Convergence Analysis Results

Our RFA-LCF is solved alternately, so we present its convergence analysis results on COIL100 and HWDB1.1-D as examples. The convergence results are shown in Fig.2.We can find that the divergence between two consecutive new representations by our RFA-LCF method is non-increasing in the iteration, and the convergence speed is also fast.

(a) COIL100 (b) HWDB1.1-D

Fig.2: Convergence curve of our RFA-LCF on two real databases.

Quantitative Evaluation of Image Clustering

**Clustering Evaluation Process and Metric.** For the quantitative evaluations, we perform the K-means clustering with cosine distance over the new representation of each method. Following the procedures in (Sugiyama et al. 2007), for each number K, we choose K categories from each dataset randomly and use the data of K categories to form the data matrix X for learning the representation for clustering. For each algorithm, the rank R is set to K+1 as [19] and we average results over 30 random initializations for K-means clustering algorithm. Accuracy (AC) and F-measure (He et al. 2006) are used as the quantitative evaluation methods.

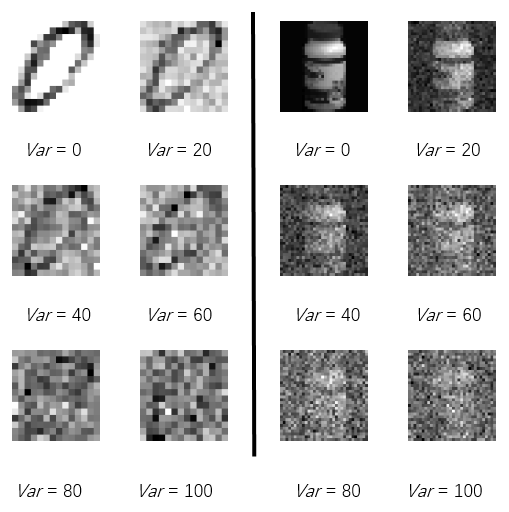
***Clustering Evaluation Results.***We evaluate each method for clustering the image data of six real-world databases, including two face databases (CMU PIE and UMIST), two object databases (COIL100 and ETH80) and two handwriting databases (HWDB1.1-D and HWDB1.1-L). For each database, we vary K from 2 to 10 with step 1, and average the results over 10 random selections of the K categories to avoid the bias. Note that the statistics including the mean and highest AC values are shown in Table II, from which we can find that our RFA-LCF delivers higher AC values than other compared methods in the investigated cases.

Clustering Image Data against Corruptions

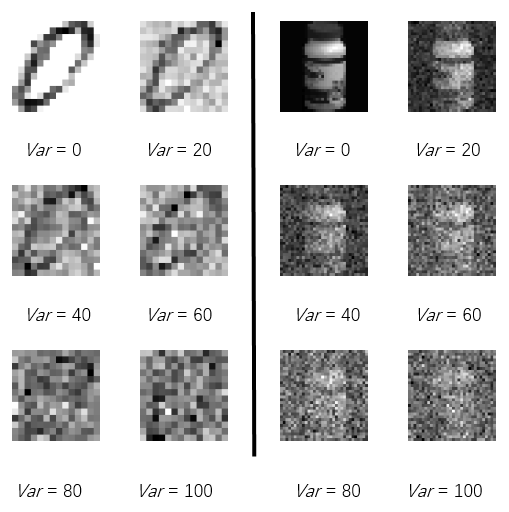
We also prepare a simulation to evaluate the performance of clustering noisy image data. HWDB1.1-D and COIL100 are selected as an example. To corrupt the data *X*, we add random Gaussian noise with the variance being 0-100 with interval 10 into the gray values of the selected image pixels. The examples of corrupted images and the clustering results in terms of F-measure by *k*-means clustering on the corrupted images are shown in Fig.3. Note that the results are obtained by choosing 2 categories and the F-measure is averaged based on 50 random selections of categories and *k*-means clustering to avoid the randomness. We can find that the result of each method generally goes down with the increasing levels of noise, i.e., the corrupted noisy data can indeed cause negative effects on the data representation and clustering. Our RFA-LCF can outperform the other methods for its more reasonable unified formulation.

**Table II.** Mean and highest clustering accuracy (AC) over the used six public image databases.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Dataset**  **Method** | **CMU PIE face database** | | **UMIST face database** | | **COIL100 object database** | |
| ***Mean±std (%)*** | ***Best (%)*** | ***Mean±std (%)*** | ***Best (%)*** | ***Mean±std (%)*** | ***Best (%)*** |
| NMF | 23.99±12.51 | 51.2 | 24.36±12.49 | 51.4 | 27.50±11.76 | 52.9 |
| GNMF | 28.10±10.96 | 52.6 | 30.50±12.39 | 58.5 | 31.84±12.09 | 59.6 |
| PNMF | 39.20±11.73 | 60.8 | 46.72±12.34 | 71.6 | 58.14±9.25 | 76.0 |
| DNMF | 31.82±9.01 | 53.5 | 30.86±14.08 | 59.3 | 63.63±13.86 | 90.9 |
| DSNMF | 31.98±11.08 | 57.5 | 38.83±12.78 | 62.5 | 61.85±16.29 | 91.1 |
| PAMGNMF | 36.81±15.53 | 63.3 | 46.04±10.86 | 64.7 | 64.71±13.14 | 89.9 |
| CF | 21.97±13.38 | 51.3 | 21.97±13.38 | 51.3 | 44.61±14.44 | 75.5 |
| LCCF | 37.80±11.25 | 61.9 | 39.01±11.96 | 62.6 | 52.78±16.58 | 85.8 |
| GRLCF | 34.94±13.65 | 62.3 | 43.29±12.14 | 66.5 | 48.03±15.33 | 79.8 |
| LGCF | 36.06±14.07 | 62.5 | 42.61±12.78 | 63.3 | 48.05±15.48 | 80.2 |
| LCF | 37.09±12.41 | 57.3 | 42.15±12.66 | 63.3 | 47.61±15.27 | 79.1 |
| GCF | 35.82±12.14 | 53.3 | 41.57±11.42 | 61.1 | 42.71±12.31 | 67.3 |
| **RFA-LCF** | **41.58±12.84** | **65.9** | **48.83±13.10** | **74.7** | **67.59±13.88** | **92.5** |
|  | **ETH80 object database** | | **HWDB1.1-D handwriting** | | **HWDB1.1-L handwriting** | |
| ***Mean±std (%)*** | ***Best (%)*** | ***Mean±std (%)*** | ***Best (%)*** | ***Mean±std (%)*** | ***Best (%)*** |
| NMF | 24.94±12.42 | 52.3 | 25.55±12.34 | 52.7 | 24.67±11.84 | 51.3 |
| GNMF | 26.42±11.60 | 52.4 | 30.64±11.18 | 54.0 | 27.74±11.40 | 53.6 |
| PNMF | 45.65±13.24 | 72.1 | 35.08±12.73 | 63.4 | 37.51±11.52 | 59.2 |
| DNMF | 22.06±13.33 | 51.2 | 27.02±11.49 | 51.7 | 25.16±11.59 | 51.3 |
| DSNMF | 26.06±12.27 | 54.2 | 33.95±10.10 | 54.7 | 27.93±12.15 | 54.3 |
| PAMGNMF | 25.49±12.41 | 52.5 | 28.73±11.63 | 53.1 | 25.39±12.33 | 52.4 |
| CF | 33.65±13.23 | 63.5 | 28.99±12.01 | 55.3 | 28.66±11.21 | 53.7 |
| LCCF | 34.97±14.58 | 67.9 | 32.74±12.58 | 58.5 | 30.62±11.47 | 54.1 |
| GRLCF | 36.39±13.61 | 66.8 | 34.07±12.59 | 60.8 | 32.79±11.72 | 57.1 |
| LGCF | 36.34±13.42 | 66.0 | 34.95±12.88 | 62.1 | 32.63±11.83 | 56.6 |
| LCF | 36.04±13.79 | 67.1 | 29.17±11.91 | 54.9 | 31.43±11.74 | 56.1 |
| GCF | 33.40±13.29 | 64.0 | 28.89±11.97 | 55.0 | 30.52±11.16 | 54.2 |
| **RFA-LCF** | **49.05±14.63** | **78.0** | **38.71±13.59** | **68.7** | **39.85±11.83** | **62.2** |

* *

**(a) HWDB1.1-D (b) Images with corrupted pixels**

* *

**(c) COIL100 (d) Images with corrupted pixels**

*Fig.3: Clustering image data against different levels of corruptions.*

Parameter Sensitivity Analysis

We explore the effects of the model parameters,andon the clustering results of RFA-LCF. F-measure is used as the evaluation metric and UMIST face database is selected. Since RFA-LCF has three parameters, we use the widely-used grid search strategy (Zhang et al. 2016), i.e., we fix one of the parameters and tune other two from the candidate set. The analysis results over 2 categories are shown in Fig.4, where the results are averaged over 30 times selections of categories and central points in *k*-means. We can find that RFA-LCF delivers stable results over a wide range of parameter settings, i.e., our RFA-LCF method is robust to the model parameters.



*Fig.4: Clustering accuracy of our RFA-LCF under various parameters, where: (left) Fix**to tune  and ; (middle) Fixto tuneand* *; (right) Fix to tune  and* *.*

Concluding Remarks

We proposed a simple yet effective robust flexible auto-weighted local-coordinate concept factorization model for unsupervised representation and clustering. Our framework aims at improving the accuracy of encoding neighborhood and factorizing data against noise and outliers by seamlessly integrates the robust flexible concept factorization, robust sparse local coordinate coding, error correction and adaptive weights learning. The derived adaptive weighting strategy can avoid the tricky process of selecting the optimal parameters in defining the affinity. The flexible residue, local-coordinate coding and adaptive weighting are also performed in the recovered clean data space for potentially enhancing the representation results for clustering.

Extensive clustering evaluation have verified the validity of our method. In future, how to use our model to involve new data still remains unclear and should be investigated. The optimal selection of the rank *R* should also be studied.

References

Jolliffe, I. 1986. Principal Component Analysis. *Springer Berlin*, 87(100):41-64.

Golub, G. H., Reinsch, C. 1970. Singular value decomposition and least squares solutions. *Numerische mathematik*, 14(5): 403-420.

Gray, R. 1984. Vector quantization. *IEEE Assp Magazine*, 1(2):4-29.

Lee, D., and Seung, H. 1999. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401: 788-791.

Wei, X., and Gong, Y. 2004. Document clustering by concept factorization. *In: Proceedings of the ACM SIGIR conference on Research and de­velopment in information retrieval*.

Yuan, Z., and Oja, E. 2010. Linear and Nonlinear Projective Nonnegative Matrix Factorization. *IEEE Trans. on Neural Networks,* 21(5):734-749.

Cai, D.; He, X.; and Han, J. 2011. Graph regularized nonnegative matrix factorization for data representation. *IEEE Trans. Pattern Anal. Mach. Intell.,* 33(8): 1548-1560.

Liu, H.; Wu, Z.; and Li, X. 2012. Constrained nonnegative matrix factorization for image representation. *IEEE Trans. on Pattern Anal. Mach. Intell.*, 34(7): 1299 -1311.

Cai, D.; He, X.; and Han, J. 2011. Locally consistent concept factorization for document clustering. *IEEE Trans. on Knowledge and Data Engineering,* 23(6): 902-913.

Liu，H.; Yang, Z.; Yang, J.; Wu, Z.; and Li, X. 2014. Local coordinate concept factorization for image representation. *IEEE Trans. on Neural Networks and Learning Systems*, 25(6).1071-1082.

Yang, Y.; Shen, H.; Ma, Z.; Huang, Z.; and Zhou, X. 2011. L2, 1-Norm Regularized Discriminative Feature Selection for Unsupervised Learning. *In: Proceeding of the International Joint Conferences on Artificial Intelligence (IJCAI)*.

Zhang, Z.; Liu, C. L.; and Zhao, M. 2015. A Sparse Projection and Low-Rank Recovery Framework for Handwriting Representation and Salient Stroke Feature Extraction. *ACM Trans. on Intelligent Systems and Technology*, 6(1). 619-634.

Nayar, S.; Nene, S.; and Murase, H. 1996. Columbia object image library (coil 100). *Department of Comp. Science, Columbia University, Tech. Rep.* CUCS-006-96.

Samaria, F., and Harter, A. 1994. Parameterisation of a stochastic model for human face identification. In: *Proceedings of IEEE Workshop on Applications of Computer Vision*, 138-142.

Lu, J.; Plataniotis, K.; and Venetsanopoulos, A. 2003. Face recognition using LDA-based algorithms. *IEEE Trans. on Neural Networks*, 14(1):195-200.

Roweis, S., and Saul, L. 2000. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323- 2326.

Tenenbaum, J.; De, S.; and Langford, J. 2000. A global geometric framework for nonlinear dimensionality reduction. *Science*, 290(5500):2319-2323.

Sugiyama, M. 2007. Dimensionality reduction of multimodal labeled data by local fisher discriminant analysis. *Journal of Machine Learning Research*, 8:1027-1061.

Zhang, Z.; Li, F.; Zhao, M.; Zhang, L.; and Yan, S. 2016. Joint Low-Rank and Sparse Principal Feature Coding for Enhanced Robust Representation and Visual Classification. *IEEE Trans. on Image Processing*, 25(6):2429-2443.

Liu, H.; Yang, G.; Wu, Z.; and Cai, D. 2014. Constrained concept factorization for image representation. *IEEE Transactions on Cybernetics*, 44(7):1214-1224, 2014.

He, X.; Cai, D.; and Niyogi, P. 2006. Laplacian score for feature selection. *In: Advances in neural information processing systems*, pp.507-514.

Belkin, M., Niyogi, P. 2002. Laplacian eigenmaps and spectral techniques for embedding and clustering. *In: Neural Infor­mation Processing Systems,* 14(6): 585-591.

Ye, J., and Jin, Z. 2017. Graph-Regularized Local Coordinate Concept Factorization for Image Representation. *Neural Processing Letters*, 46(2):427- 449.

Cormen, T. H. 2009. Introduction to algorithms. *MIT press*.

Meng, Y.; Shang, R.; Jiao, L.; Zhang, W.; Yuan, Y.; and Yang, S. 2018. Feature Selection Based Dual-graph Sparse Non-negative Matrix Factorization for Local Discriminative Clustering. *Neurocomputing*, 290:87-99.

Wechsler, H.; Phillips, J. P.; Bruce, V.; Soulie, F. F.; and Huang, T. S. 2012. Face recognition: From theory to applications. *Springer Science & Business Media*, vol.163.

Sim, T.; Baker, S.; and Bsat, M. 2002. The CMU pose, illumination, and expression (PIE) database. In: *IEEE International Conference on Automatic Face and Gesture Recognition,* pp.53-58.

Zhang, Z.; Jiang, W.; Qin, J.; Zhang, L.; Li, F.; Zhang, M.; and Yan, S. 2017. Jointly Learning Structured Analysis Discriminative Dictionary and Analysis Multiclass Classifier. *IEEE Trans. on Neural Networks and Learning Systems*.

Ye, J., and Jin, Z. 2014. Dual-graph regularized concept factorization for clustering. *Neurocomputing*, 138(11):120-130.

Shang, F.; Jiao, L.C.; and Wang, F. 2012. Graph dual regularization non-negative matrix factorization for co-clustering. *Pattern Recognition*. 45(6):2237-2250.

Nie, F.; Xu, D.; Tsang, W.; and Zhang, C. 2010. Flexible Manifold Embedding: A Framework for Semi-supervised and Unsupervised Dimension Reduction. *IEEE Transactions on Image Processing*, 19(7):1921-1932, 2010.

Li, H.; Zhang, J.; and Liu, J. 2017. Graph-regularized CF with local coordinate for image representation. *Journal of Visual Communication & Image Representation*.

Meng, Y.; Shang, R.; Jiao, L.; Zhang, W.; Yuan, Y.; and Yang, S. 2018. Feature Selection Based Dual-graph Sparse Non-negative Matrix Factorization for Local Discriminative Clustering. *Neurocomputing*, 290:87-99.

Shu, Z.; Wu, X.; Fan, H.; Huang, P.; Wu, D.; Hu, C.; and Ye, F. 2017. Parameter-less Auto-weighted Multiple Graph regularized Nonnegative Matrix Factorization for Data Representation. *Knowledge-Based Systems*, 131:105-112.

Nie, F.; Wang, X.; Jordan, M.; and Huang, H. 2016. The Constrained Laplacian Rank algorithm for graph-based clustering. In: *Proceeding of AAAI Conference on Artificial Intelligence*.

1. Copyright © 2018, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved. [↑](#footnote-ref-2)