Applied Intelligence (2022) 52:7653–7669
https://doi.org/10.1007/s10489-021-02826-0

Correntropy-based dual graph regularized nonnegative matrix factorization with L_p smoothness for data representation

Zhenqiu Shu[1] · Zonghui Weng[2] · Zhengtao Yu[1] · Congzhe You[2] · Zhen Liu[3] · Songze Tang[4] · Xiaojun Wu[3]

Accepted: 8 September 2021 / Published online: 7 October 2021

Abstract
Nonnegative matrix factorization methods have been widely used in many applications in recent years. However, the clustering performances of such methods may deteriorate dramatically in the presence of non-Gaussian noise or outliers. To overcome this problem, in this paper, we propose correntropy-based dual graph regularized NMF with L_P smoothness (CDNMFS) for data representation. Specifically, we employ correntropy instead of the Euclidean norm to measure the incurred reconstruction error. Furthermore, we explore the geometric structures of both the input data and the feature space and impose an L_p norm constraint to obtain an accurate solution. In addition, we introduce an efficient optimization scheme for the proposed model and present its convergence analysis. Experimental results on several image datasets demonstrate the superiority of the proposed CDNMFS method.

Keywords NMF · Correntropy · Smoothness · L_p norm · Dual graph · Geometric structures · Convergence

1 Introduction

With the development of science and technology, massive amounts of high-dimensional data are produced in the real world. Therefore, it is still a foundational topic to explore the intrinsic low-dimensional representations derived from such high-dimensional data [1–3]. In the past few decades, various data representation methods based on matrix factorization have been proposed due to their efficiency and effectiveness. Until now, they have achieved encouraging performances in many real applications, such as object classification [4, 5], image clustering and recognition [6–8], and document analysis [9, 10].

Nonnegative matrix factorization (NMF) [11], as a well-known part-based representation method, has received extensive attention because of its strong interpretability and clear physical significance [12–14]. NMF aims to find two low-dimensional nonnegative matrices and then uses their product to optimally approximate the original matrix. Recently, some variants of NMF have been proposed by using different regularization constraints. Cai et al. [8] proposed a graph regularized NMF (GNMF) method to discover the local manifold structure of data by constructing a graph regularizer. Leng et al. [15] further proposed a GNMF approach incorporating the L_p smoothness method by imposing an additional L_p norm constraint. Chen et al. [16] proposed a constrained NMF (CNMF) method, which constructs a restricted matrix to extract image concepts consistent with known label information. Thus, it ensures that data points sharing the same label are mapped to the same concept. Guan et al. [17] presented a manifold regularized discriminative NMF (MD-NMF) method. It fully takes the manifold structure embedded in data into account and maximizes the margins between the low-dimensional representation of data from different categories, simultaneously. Shang et al. [18] proposed a dual graph regularized NMF (DNMF) method for co-clustering. In DNMF, the local geometric structures of both the data and feature space are effectively exploited by constructing two graph regularizers.

✉ Zhenqiu Shu
shuzhenqiu@163.com

[1] Faculty of Information Engineering and Automation, Kunming University of Science and Technology, Kunming 650500, China

[2] School of Computer Engineering, Jiangsu University of Technology, Changzhou 231001, China

[3] Jiangsu Provincial Engineering Laboratory of Pattern Recognition and Computational Intelligence, Jiangnan University, Wuxi 231001, China

[4] Department of Criminal Science and Technology, Nanjing Forest Police College, Nanjing 210023, China

All the aforementioned methods measure the reconstruction error by adopting the Euclidean norm, which is comprehensible and mathematically processable under the Gaussian assumption. However, they are sensitive to non-Gaussian noise or outliers in the observation data. To improve the robustness of data representation, He et al. [19] applied the maximum correntropy criterion to sparse representation and then adopted the half-quadratic technique to optimize the proposed model. Zhou et al. [20] proposed a robust semi-supervised concept factorization algorithm, which can eliminate the negative effects of the non-Gaussian noise by using the maximum correntropy criterion. Wang. et al. [21] proposed a correntropy induced metric-based graph regularized NMF (CGNMF) algorithm, and experimental results have been demonstrated its robustness and effectiveness with real datasets.

In this paper, we propose correntropy-based dual graph regularized NMF with L_P norm smoothness (CDNMFS) method for data representation. Specifically, the proposed CDNMFS method employs the correntropy to measure the incurred approximation error. In addition, it explores the manifold structure embedded in dual spaces using the regularization technology, and imposes the L_p norm constraint on the basis matrix. Experimental results on real-world datasets show that the proposed CDNMFS method can effectively enhance the effectiveness of the obtained representation.

The main contributions of this work are summarized as follows:

(1) Our proposed method adopts the correntropy as the measure and thus can shows more robustness to non-Gaussian noise and outliers than other methods using the Euclidean distance.
(2) A dual graph regularizer is constructed in CDNMFS to preserve the manifold structures of both data and features. In addition, we can achieve a more accurate solution by enforcing the *Lp* norm constraint.
(3) We provide an efficient optimization scheme to solve the proposed model and its convergence analysis is also given in this work. The experimental results in clustering tasks show the superiority of the proposed method.

The remainder of this paper is organized as follows: Sect. 2 briefly reviews the related works. Section 3 introduces the proposed CDNMFS algorithm. Section 4 presents the experimental results, and Sect. 5 concludes this work.

2 The related works

In the next four subsections, we briefly review the previous researches of the NMF model.

2.1 NMF

NMF represents an original data matrix $X = [x_{ij}] \in \mathbb{R}^{M\times N}$ by the product of two nonnegative low-dimensional matrices $U = [u_{ik}] \in \mathbb{R}^{M\times K}$ and $V = [v_{jk}] \in \mathbb{R}^{N\times K}$. Therefore, NMF minimizes the objective function as follows:

$$O_{NMF} = \sum_{i=1}^{N}\sum_{j=1}^{M}(X_{ij} - (UV^T)_{ij})^2 = \|X - UV^T\|_F^2, \quad s.t.\ U \geq 0, V \geq 0 \tag{1}$$

where $\|\cdot\|_F$ represents the Frobenius norm. Using the multiplicative iterative algorithm proposed by Lee and Seung [22], the updating rules of NMF can be obtained as follows:

$$u_{ik} \leftarrow u_{ij}\frac{(XV)_{ij}}{(UV^TV)_{ij}}, \tag{2}$$

$$v_{jk} \leftarrow v_{ij}\frac{(X^TU)_{ij}}{(VU^TU)_{ij}}. \tag{3}$$

2.2 GNMF

Motivated by manifold learning theory, GNMF aims to embed the geometric manifold structure of the original data into a low-dimensional representation during matrix decomposition. Specifically, it constructs a nearest neighbor graph to approximately model the intrinsic local geometry of the input data, and then integrates this graph into the NMF model by adopting the regularization technology. Therefore, the objective function of GNMF is given as follows:

$$\text{OGNMF} = \|X - UV^T\|_F^2 + \lambda Tr(V^TLV), \quad s.t.\ U \geq 0, V \geq 0, \tag{4}$$

where λ is the regularization parameter. $L = D - W$ is a graph Laplacian matrix, where W is an affinity matrix and D is a diagonal matrix, $D_{jj} = \sum_l W_{jl}$.

By employing a similar optimization scheme, the updating rules of GNMF can be derived as follows:

$$u_{ij}^{t+1} \leftarrow u_{ij}^{t}\frac{(XV)_{ij}}{(UV^TV)_{ij}}, \tag{5}$$

$$v_{ij}^{t+1} \leftarrow v_{ij}^{t}\frac{(X^TU + \lambda WV)_{ij}}{(VU^TU + \lambda DV)_{ij}}. \tag{6}$$

2.3 CGNMF

Wang et al. [21] proposed a CGNMF method that not only effectively reduces the influence of non-Gaussian noise and outliers by minimizing the correntropy between the data matrix and its reconstruction but also preserves the intrinsic geometric structure of data using graph regularization. The cost function of CGNMF is formulated as follows:

$$\min_{U\geq 0,V\geq 0} CIM(X,UV)+\frac{\lambda}{2}tr(VLV^T), \tag{7}$$

where λ is a nonnegative parameter. In Eq. (7), the first term employs the correntropy metric to measure the reconstruction cost, and the second term stands for the graph regularization term.

2.4 DNMF

Shang et al. [18] proposed the DNMF method for co-clustering, noting that the data and features lie on two nonlinear low-dimensional manifolds. Thus, DNMF explores the two manifolds by constructing two nearest neighbor graphs, respectively. The DNMF model can be formulated as

$$O_{DNMF}=\begin{matrix}\left\|X-UV^T\right\|_F^2+\lambda tr(V^TL_VV)+\mu(U^TL_UU)\\ s.t.\ U\geq 0, V\geq 0\end{matrix}, \tag{8}$$

where λ and μ are two regularization parameters. $L_U=D_U-W_U$ is the Laplacian matrix of the feature graph, and $L_V=D_V-W_V$ is the Laplacian matrix of the data graph.

Similarly, the optimization problem of DNMF can be solved by the following updating rules:

$$U_{ij}\leftarrow U_{ij}\frac{\left[XV+\mu W_UU\right]_{ij}}{\left[UV^TV+\mu D_UU\right]_{ij}}, \tag{9}$$

$$V_{kj}\leftarrow V_{kj}\frac{\left[X^TU+\lambda W_V\right]_{kj}}{\left[VU^TU+\lambda D_VV\right]_{kj}}. \tag{10}$$

3 CDNMFS

In this section, we introduce the proposed CDNMFS method in detail.

3.1 Motivation

Most existing NMF-based algorithms use the Euclidean norm to measure the approximation error and thus their performances usually drop sharply when dealing with the data containing non-Gaussian noise or outliers. To address this problem, we propose the CDNMFS method for dealing with complex high-dimensional data. Therefore, we apply the correntropy instead of the Euclidean norm to overcome this limitation. In addition, our proposed CDNMFS method discovers the geometric structures of both the data manifold and feature manifold by constructing a dual regularizer. Furthermore, we impose L_P norm constraint on the basis matrix to achieve an accurate solution.

First, we will introduce the definition of entropy.

3.2 Correntropy

Recently, correntropy has been widely studied in the fields of machine learning and computer vision [23–28]. One of its important advantages is that it can effectively deal with data containing with non-Gaussian noise or outliers using the information theoretical learning. Correntropy is used to calculate the nonlinear similarity of two arbitrary variables x and y as follows:

$$C(X,Y)=E[k(X,Y)]=\int k(x,y)dF_{XY}(x,y), \tag{11}$$

where $E[\cdot]$ denotes the mathematical expectation operator and $k(\cdot,\cdot)$ is kernel function satisfying the Mercer theory. $F_{XY}(x,y)$ stands for the joint distribution function of (X, Y). Correntropy maps the nonlinear input data into a higher dimensional space using kernel function.

The kernel $K(\cdot,\cdot)$ employs the Gaussian kernel, and thus its definition can be given as follows:

$$k_\sigma(x,y)=g(x-y)=\exp\left(-\frac{(x-y)^2}{2\sigma^2}\right), \tag{12}$$

where σ is the kernel bandwidth parameter. Since this parameter is unknown for the joint distribution of two random variables, the number of available samples is limited. The correntropy $\widehat{C}_\sigma(x,y)$ can be calculated by the following formulation:

$$\widehat{C}_\sigma(x,y)=\frac{1}{N}\sum_{n=1}^{N}g(x_n-y_n). \tag{13}$$

Compared with the Euclidean norm, maximum correntropy criterion (MCC) not only utilizes a higher-order relationship, but also exhibits more robustness to

non-Gaussian noise and outliers. Therefore, the objective function of the MCC-based NMF is defined as

$$\begin{aligned}&\max_{\substack{U,V\\ s.t.U\geq 0,V\geq 0}} F(U,V)\\ &F(U,V)=\sum_{i=1}^{N}k_\sigma(e_\sigma)\\ &=\sum_{i=1}^{N}k_\sigma(\sqrt{\sum_{j=1}^{M}(x_{ij}-\sum_{k=1}^{K}u_{ik}v_{kj})^2})\\ &=\sum_{i=1}^{N}\exp(-\frac{\sum_{j=1}^{M}(x_{ij}-\sum_{k=1}^{K}u_{ik}v_{kj})^2}{2\sigma^2})\end{aligned}. \tag{14}$$

where $e_\sigma=\sqrt{\sum_{j=1}^{M}(x_{ij}-\sum_{k=1}^{K}u_{ik}v_{kj})}$.

3.3 Objective function of the proposed method

The proposed CDNMFS method aims to maximize the objective function as follows:

$$\begin{aligned}\max_{u,v} O_{\partial CDNMFS}=&\sum_{i=1}^{M}g\left(\sqrt{\sum_{j=1}^{N}\left(x_{ij}-\left(UV^T\right)_{ij}\right)^2-\lambda Tr\left(V^TL_VV\right)}\right)\\ &-\alpha Tr\left(\pi U^TL_UU\right)-2\mu\|U\|^P\end{aligned} \tag{15}$$

where $\lambda\geq 0$,$\alpha\geq 0$ and $\mu\geq 0$ are the regularization parameters. Generally, the value of P is empirically set between 1 and 2. L_V and L_U are the Laplacian matrices of the data graph and the feature graph, respectively.

3.4 Optimization algorithm

The proposed CDNMFS method employs correntropy instead of the traditional Euclidean distance as the measurement metric. Notably, the proposed model is a nonconvex problem and thus is difficult to optimize in practice. Fortunately, we can transform the CDNMFS model into a quadratic function with a multiplicative form by introducing the half-quadratic (HD) technique based on conjugate function theory. Using the property of the convex conjugate function, we state proposition 1 as follows [29]:

Proposition 1: *A convex conjugate function $\phi(\cdot)$ of $g(x)$ existes as*

$$g(x)=\max_z(z\frac{\|x\|^2}{\sigma^2}-\phi(z)), \tag{16}$$

where z is a scalar variable. The maximum value for a fixed x is reached at $z=-g(x)$. By substituting the Eq. (16) into the Eq. (15), the objective function of CDNMFS can be reformulated as follows:

$$\begin{aligned}\max_{U,V,z} O_{CDNMFS}=&\sum_{i=1}^{M}g\left(\frac{z_i}{\sigma^2}\left(\sqrt{\sum_{j=1}^{N}\left(x_{i,j}-\left(UV^T\right)_{i,j}\right)^2}\right)\right)-\lambda Tr\left(V^TL_VV\right)\\ &-\alpha Tr\left(\pi U^TL_UU\right)-2\mu\|U\|^P\end{aligned} \tag{17}$$

where $z=\left[z_1,\cdots,z_M\right]^T$ is an auxiliary vector. By fixing U and V, we maximize Eq. (17) with respect to z as follows:

$$\begin{aligned}z_i&=-g(\sqrt{\sum_{j=1}^{N}(X_{ij}-(UV^T)_{ij})^2}\\ &=-\exp(-\frac{\sum_{j=1}^{N}(x_{ij}-(UV^T)_{ij})^2}{2\sigma^2})\end{aligned}, \tag{18}$$

and the value of the kernel bandwidth parameter σ is empirically set to

$$\sigma=\sqrt{\frac{1}{M}\sum_{i=1}^{M}\sum_{j=1}^{N}(X_{i,j}-(UV^T)_{i,j})^2}. \tag{19}$$

By fixing σ, the objective function of O_{CDNMFS} in Eq. (17) is reformulated as follows:

$$\begin{aligned}\max_{U,V} O_{CDNMFS}=&\sum_{i=1}^{M}(\frac{z_i}{\sigma^2}(\sum_{j=1}^{N}(X_{i,j}-(UV^T)_{i,j})^2))-\lambda Tr(V^TL_VV)\\ &-\alpha Tr(U^TL_UU)-2\mu\|U\|^P\end{aligned}. \tag{20}$$

To simplify the calculation, Eq. (20) is rewritten as the following equivalent optimization problem:

$$\begin{aligned}\min_{U,V} O_{CDNMFS}=&\sum_{i=1}^{M}(-\frac{z_i}{\sigma^2}(\sum_{j=1}^{N}(X_{i,j}-(UV^T)_{i,j})^2))+\lambda Tr(V^TL_VV)\\ &+\alpha Tr(U^TL_UU)+2\mu\|U\|^P\end{aligned}. \tag{21}$$

Equation (21) can be further reformulated as follows:

$$\begin{aligned}f&=Tr((X-UV^T)^TQ(X-UV))+\lambda Tr(V^TL_VV)+\alpha Tr(U^TL_UU)+2\mu\|U\|^P\\ &=Tr(X^TQX)-2Tr(VU^TQX)+Tr(VU^TQUV^T)+\lambda Tr(V^TL_VV)\\ &\quad+\alpha Tr(U^TL_UU)+2\mu\|U\|^P\end{aligned}, \tag{22}$$

where Q is a diagonal matrix with

$$Q_{ii}=-\frac{z_i}{\sigma^2}=\sigma^{-2}\exp(-\frac{1}{2\sigma^2}\sum_{j=1}^{N}(X_{i,j}-(UV^T)_{i,j})^2). \tag{23}$$

Similarly, the multiplicative iterative algorithm is applied to solve Eq. (23). Since $U\geq 0, V\geq 0$, $\Psi=\left[\psi_{ik}\right]$ and $\phi=\left[\varphi_{jk}\right]$ denote the Lagrange multipliers for U and Z, respectively. Therefore, the Lagrange function L of Eq. (23) can be written as follows:

$$\begin{aligned}L&=Tr((X-UV^T)^TQ(X-UV))+\lambda Tr(V^TL_VV)+\alpha Tr(U^TL_UU)+2\mu\|U\|^P\\ &=Tr(X^TQX)-2Tr(VU^TQX)+Tr(VU^TQUV^T)+\lambda Tr(V^TL_VV)\\ &\quad+\alpha Tr(U^TL_UU)+Tr(\psi U^T)+Tr(\varphi V^T)+2\mu\|U\|^P\end{aligned}, \tag{24}$$

Then the partial derivatives of L with respect to U and Z are given as follows:

$$\frac{\partial L}{\partial U} = -2QXV + 2QUV^TV + 2\alpha L_U U + \psi + 2\mu PU^{P-1}, \tag{25}$$

$$\frac{\partial L}{\partial V} = -2X^TQU + 2VU^TQU + 2\lambda L_V V + \varphi. \tag{26}$$

According to the Karush–Kuhn–Tucker (KKT) conditions $\varphi U = 0$ and $\phi V = 0$, the update rules of CSNMF can be derived as

$$u_{ik} \leftarrow u_{ik}\frac{(QXV + \alpha W_U U)_{ik}}{(QUV^TV + \alpha D_U U + P\mu U^{P-1})_{ik}}, \tag{27}$$

$$v_{jk} \leftarrow v_{jk}\frac{(X^TQU + \lambda W_V V)_{jk}}{(VU^TQU + \lambda D_V V)_{jk}}. \tag{28}$$

The detailed steps of the proposed CDNMFS method are summarized in Algorithm 1.

Algorithm 1 CDNMFS algorithm Input: Data matrix:$X \in R^{m\times n}$; Parameters:λ, α, μ, P\lambda ,\alpha ,\mu ,P

Output: U and V;

1. Randomly initialize $U \geq 0$ and $V \geq 0$;
2. W**hile** not converged
   (1) Update σ by using Eq. (19);
   (2) Update Q by using Eq. (23);
   (3) Update U by using Eq. (27);
   (4) Update V by using Eq. (28);

End

3.5 Convergence analysis

In this subsection, we provide the convergence proof for the proposed CDNMFS algorithm by using the derived updating rules. First, we introduce some related definitions and lemmas.

Definition 1: *$G(x, x')$ is called as an auxiliary function of $F(x)$, if $G(x, x')$ satisfies the following conditions:*

$$G(x, x') \geq F(x), G(x, x) = F(x) \tag{29}$$

Lemma 1: *If G is an auxiliary function of F, F is made nonincreasing by adopting the following updating rule.*

$$x^{t+1} = \arg\min_x G(x, x^t). \tag{30}$$

Proof: According to Definition 1, we know that

$$F(x^{t+1}) \leq G(x^{t+1}, x^t) \leq G(x^t, x^t) = F(x^t), \tag{31}$$

and thus we have

$$F\left(x^{t+1}\right) \leq F\left(x^t\right). \tag{32}$$

$F_{U_{ab}}$ denotes the part of the objective function O_{CDNMFS} that is relevant only to element U_{ab} in U. Therefore, we have

$$F'_{u_{ab}} = \left(\frac{\partial O_{CDNMFS}}{\partial U}\right)_{ab} = (-2QXV + 2QUV^TV + 2\alpha L_U U + 2\mu PU^{P-1})_{ik}, \tag{33}$$

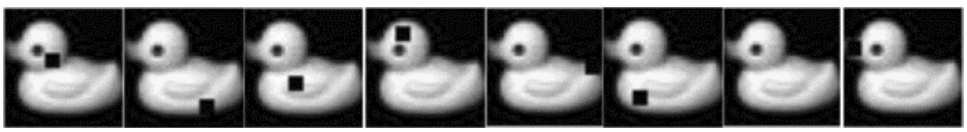


Fig. 1 Some samples from the COIL20 dataset contaminated by 4×4 black blocks

Table 1 AC on the COIL20 dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
10	0.647	0.674	0.528	0.810	0.659	0.814	0.805	0.710	**0.829**
11	0.651	0.645	0.515	0.813	0.645	0.811	0.809	0.746	**0.853**
12	0.672	0.654	0.558	0.800	0.647	0.808	0.802	0.750	**0.831**
13	0.680	0.648	0.586	0.818	0.655	0.838	0.832	0.747	**0.855**
14	0.672	0.644	0.580	0.819	0.684	0.818	0.814	0.764	**0.836**
15	0.657	0.648	0.556	0.817	0.654	0.800	0.796	0.763	**0.825**
16	0.682	0.646	0.571	0.822	0.674	0.827	0.832	0.772	**0.843**
17	0.654	0.643	0.585	0.807	0.662	0.807	0.819	0.782	**0.820**
18	0.674	0.642	0.577	0.815	0.637	0.810	0.800	0.754	**0.812**
19	0.621	0.639	0.584	0.778	0.640	0.755	0.777	0.715	**0.795**
20	0.643	0.601	0.597	0.778	0.650	0.768	0.765	0.763	**0.766**
AVG	0.659	0.644	0.567	0.807	0.655	0.805	0.805	0.751	**0.824**

The bold values mean the best performance in all methods

$$F''_{u_{ik}} = (\frac{\partial^2 O_{CDNMFS}}{\partial U^2})_{ik} = 2Q_{ii}(V^T V)_{kk} + 2(\alpha L_U)_{ii} + 2\mu P(P-1)(U^{P-2})_{ik}, \tag{34}$$

$$F'_{V_{jk}} = (\frac{\partial O_{CDNMFS}}{\partial V})_{jk} = (-2X^T QU + 2VU^T QU + 2\alpha L_U U + 2\lambda L_V V)_{jk}, \tag{35}$$

$$F''_{V_{jk}} = (\frac{\partial^2 O_{CDNMFS}}{\partial V^2})_{jk} = 2(U^T QU)_{kk} + 2\lambda(L_V)_{jj}. \tag{36}$$

Lemma 2: *The function*

$$G(u, u^t_{ik}) = F_{u_{ik}}(u^t_{ik}) + F'_{u_\partial}(u^t_{ik})(u - u^t_{ik}) + \frac{(QUV^T V)_{ik} + \alpha(D_U U)_{ik} + \mu P(U^{P-1})_{ik}}{u^t_{ik}}(u - u^t_{ik})^2 \tag{37}$$

is an auxiliary function of $F_{u_{ik}}$.

Proof:

Since $G(u,u) = F_{u_{ik}}(u)$ *is obvious, we only need to prove that* $G(u, u^t_{ik}) \geq F_{u_{ik}}(u)$. *To achieve this goal, the Taylor series of* $F_{u_{ik}}(u)$ *can be expanded as follows*:

$$\begin{aligned} F_{u_{ik}}(u) &= F_{u_{ik}}(u^t_{ik}) + F'_{u_{ik}}(u^t_{ik})(u - u^t_{ik}) + \tfrac{1}{2}F''_{u_{ik}}(u^t_{ik})(u - u^t_{ik})^2 \\ &= F_{u_{ik}}(u^t_{ik}) + F'_{u_{ik}}(u^t_{ik})(u - u^t_{ik}) + (u - u^t_{ik})^2 Q_{ii} \\ &[(V^T V)_{kk} + \alpha(L_u)_{ii} + \mu P(P-1)(U^{P-2})_{ik}]. \end{aligned} \tag{38}$$

By comparing Eq. (38) *and Eq.* (37), *it can be found that* $G(u, u^t_{ik}) \geq F_{u_{ik}}(u)$ *is equivalent to*

$$\frac{[(QUV^T V) + \alpha D^U U + \mu P(U^{P-1})]_{ik}}{u^t_{ik}} \geq Q_{ii}(V^T V)_{kk} + \alpha(L_U)_{ik} + \mu P(P-1)(U^{P-2})_{ik}. \tag{39}$$

We have

$$(QUV^T V)_{ik} = \sum_{l=1}^{k} u^t_{il} Q_{ii}(UV^T V)_{lb} \geq u^t_{ik} Q_{ii}(V^T V)_{kk}. \tag{40}$$

The second term is obvious, and thus (39) *holds*.

According to the above two updating rules, we can demonstrate Theorem 1.

Theorem 1: *The objective function* O_{CDNMFS} *is nonincreasing under update rules* (27) *and* (28).

Proof:

Replacing $G(u, u^t_{ik})$ *in* (30) *by* (37), *we obtain*.

$$\begin{aligned} u^{t+1}_{ik} &= u^t_{ik} - u^t_{ik} \frac{F'_{ik}(u^t_{ik})}{2(QUV^T V + \alpha D_U U + \mu P U^{P-1})_{ik}} \\ &= u^t_{ik} \frac{(QXV + \alpha W_U U)_{ik}}{(QUV^T V + \alpha D_U U + P\mu U^{P-1})_{ik}}. \end{aligned} \tag{41}$$

Similarly, we can derive that the objective function O_{CDNMFS} is nonincreasing under the updating rule in (28). Therefore, the convergence of the proposed optimization scheme can be guaranteed by Theorem 1, and thus we can achieve a local optimum of the proposed model by using these updating rules.

3.6 Computational complexity analysis

In this subsection, we discuss the computational complexity of the proposed CDNMFS algorithm. O notation is used to describe the computational cost. First, CDNMFS needs $O(N^2M + NM^2)$ to construct two k-nearest neighbor graphs. where M and N are the numbers of features and samples, respectively. In addition, CDNMFS requires $O(MNK)$ to update the kernel bandwidth parameter σ and the diagonal matrix Q, where K is the number of clusters. The computational complexity of update rules (27) and (28) is $O(MNK)$. Therefore, the total computational complexity of CDNMFS is given as follows:

$$O(tMNK + N^2M + NM^2),$$

where t denotes the number of iterations.

4 Experiments

In this section, we evaluate the performance of our proposed CDNMFS method on different real-world datasets, such as the PIE, COIL20 and YaleB datasets. We compare the proposed CDNMFS method with other state-of-the-art methods including k-means (KM), NMF, CF, GNMF, constrained NMF (CNMF) [7], CGNMF, DGNMF and deep graph regularized semi-NMF (DGSNMF)[30]. In addition, we employ accuracy (AC), normalized mutual information (NMI), precision and recall as the metrics to measure the clustering performances of these data representation methods. For detailed definition of these metrics, please refer to reference [18].

4.1 Experimental results

(1) COIL20 dataset

The COIL20 dataset contains 1440 images of 20 different objects. All the samples were resized as 32×32 pixel gray images, and thus the total COIL20 dataset could be represented by a 1024×1440 matrix. To verify the robustness of the proposed algorithm, we randomly added 4×4 black block noise to each sample. Figure 1 shows some samples contaminated by 4×4 black blocks from the COIL20 dataset contaminated by 4×4 black blocks.

In the COIL dataset, K samples were randomly selected from each category to verify the proposed

Table 2 NMI on COIL20 dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
10	0.695	0.702	0.566	0.860	0.690	0.860	0.861	0.767	**0.866**
11	0.712	0.689	0.566	0.867	0.689	0.864	0.864	0.796	**0.865**
12	0.727	0.696	0.622	0.853	0.692	0.863	0.861	0.801	**0.889**
13	0.733	0.713	0.645	0.882	0.718	0.883	0.878	0.813	**0.898**
14	0.741	0.721	0.649	0.886	0.741	0.883	0.881	0.824	**0.901**
15	0.737	0.720	0.645	0.878	0.711	0.872	0.871	0.838	**0.898**
16	0.753	0.730	0.657	0.888	0.736	0.892	0.892	0.843	**0.915**
17	0.740	0.722	0.668	0.881	0.736	0.880	0.887	0.843	**0.927**
18	0.752	0.763	0.687	0.888	0.732	0.886	0.880	0.833	**0.905**
19	0.730	0.728	0.694	0.880	0.727	0.872	0.880	0.807	**0.883**
20	0.734	0.713	0.683	0.871	0.710	0.877	0.877	0.840	**0.877**
AVG	0.732	0.718	0.643	0.876	0.716	0.876	0.876	0.817	**0.893**

The bold values mean the best performance in all methods

Table 3 Precision on COIL20 dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
10	0.561	0.561	0.284	0.742	0.568	0.739	0.744	0.576	**0.776**
11	0.571	0.528	0.298	0.748	0.558	0.743	0.740	0.634	**0.791**
12	0.588	0.540	0.322	0.710	0.546	0.729	0.724	0.614	**0.756**
13	0.587	0.542	0.361	0.750	0.555	0.764	0.754	0.632	**0.782**
14	0.578	0.543	0.332	0.757	0.586	0.755	0.741	0.644	**0.762**
15	0.566	0.539	0.332	0.729	0.542	0.727	0.713	0.667	**0.751**
16	0.588	0.552	0.334	0.742	0.573	0.750	0.759	0.674	**0.787**
17	0.542	0.531	0.386	0.724	0.571	0.713	0.734	0.669	**0.770**
18	0.573	0.559	0.397	0.741	0.559	0.730	0.712	0.638	**0.727**
19	0.531	0.538	0.412	0.697	0.548	0.668	0.694	0.560	**0.714**
20	0.536	0.510	0.45	0.719	0.518	0.696	0.696	0.641	**0.696**
AVG	0.565	0.540	0.355	0.733	0.557	0.738	0.728	0.632	**0.756**

The bold values mean the best performance in all methods

Table 4 Recall on COIL20 dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
10	0.674	0.667	0.637	0.838	0.654	0.843	0.843	0.794	**0.850**
11	0.657	0.666	0.597	0.822	0.631	0.825	0.826	0.781	**0.825**
12	0.690	0.690	0.620	0.860	0.679	0.853	0.860	0.803	**0.860**
13	0.638	0.619	0.555	0.813	0.614	0.811	0.810	0.769	**0.834**
14	0.671	0.669	0.559	0.847	0.665	0.825	0.833	0.786	**0.841**
15	0.672	0.658	0.547	0.848	0.656	0.853	0.853	0.772	**0.855**
16	0.675	0.659	0.528	0.857	0.642	0.859	0.856	0.797	**0.870**
17	0.633	0.612	0.474	0.825	0.608	0.827	0.825	0.770	**0.861**
18	0.672	0.676	0.617	0.836	0.661	0.833	0.837	0.812	**0.854**
19	0.635	0.616	0.499	0.838	0.627	0.839	0.838	0.772	**0.853**
20	0.684	0.675	0.573	0.854	0.662	0.852	0.855	0.810	**0.858**
AVG	0.664	0.655	0.564	0.840	0.645	0.838	0.840	0.788	**0.851**

The bold values mean the best performance in all methods

method. All data representation methods were run ten times and their average performances were reported as the final results. Tables 1, 2, 3 and 4 show the clustering results of all methods on the COIL dataset. It can be seen that the proposed CDNMFS method exhibited better performances than the other competitors. This is because our proposed CDNMFS method not only took the dual manifold structure of the data and the smoothness of the solution into account, but also used the cor-

rentropy as the measurement criterion to improve its robustness.

(2) PIE dataset

The PIE face dataset includes 41,368 multiposture, light, and expression facial images from 68 individuals. Each person is associated 13 different poses, 43 different lighting conditions and four different expressions. Similarly, the sizes of all sample images are 32×32 pixels, and each image can be expressed by a 1024-dimensional vector. Figure 2 shows some samples from the PIE dataset contaminated by 4×4 black blocks.

For this dataset, we adopted a similar experimental scheme as that used above. Therefore, we randomly chose K categorical samples to carry out the clustering experiments. All methods were run ten times and their average performances were recorded. Tables 5, 6, 7 and 8 summarize the clustering results of all methods on the PIE dataset. We can see that the average performance of our proposed CDNMFS algorithm outperformed the other data representation methods in terms of the AC, NMI, precision and recall metrics. Therefore, it implies that our proposed CDNMFS method can more effectively discover the intrinsic semantic structural information embedded in data than other state-of-the-art methods.

(3) YaleB dataset

Fig. 2 Some samples from the PIE dataset contaminated by 4×4 black blocks

Table 5 AC on the PIE dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
20	0.379	0.568	0.611	0.787	0.459	0.788	0.803	**0.830**	0.821
22	0.385	0.577	0.434	0.766	0.478	0.772	0.772	0.807	**0.816**
24	0.365	0.527	0.402	0.756	0.473	0.782	0.771	0.806	**0.821**
26	0.378	0.523	0.383	0.777	0.459	0.748	0.754	0.785	**0.813**
28	0.363	0.511	0.347	0.756	0.447	0.749	0.730	0.779	**0.807**
30	0.371	0.490	0.341	0.720	0.429	0.721	0.744	0.767	**0.805**
32	0.368	0.486	0.321	0.733	0.452	0.742	0.728	0.779	**0.800**
34	0.368	0.476	0.327	0.738	0.451	0.739	0.744	0.778	**0.807**
36	0.383	0.455	0.309	0.734	0.442	0.747	0.744	0.789	**0.807**
38	0.375	0.461	0.447	0.723	0.446	0.715	0.720	0.766	**0.775**
40	0.355	0.458	0.434	0.728	0.454	0.726	0.730	0.769	**0.756**
AVG	0.372	0.503	0.396	0.747	0.454	0.748	0.749	0.787	**0.802**

The bold values mean the best performance in all methods

Table 6 NMI on the PIE dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
20	0.590	0.746	0.747	0.896	0.666	0.897	0.900	0.917	**0.925**
22	0.593	0.748	0.599	0.888	0.677	0.890	0.891	0.905	**0.924**
24	0.585	0.728	0.589	0.880	0.687	0.888	0.885	0.915	**0.924**
26	0.604	0.733	0.572	0.895	0.684	0.887	0.890	0.907	**0.917**
28	0.545	0.728	0.528	0.885	0.680	0.886	0.879	0.905	**0.913**
30	0.604	0.721	0.548	0.867	0.670	0.870	0.877	0.894	**0.914**
32	0.607	0.718	0.538	0.888	0.694	0.890	0.884	0.915	**0.925**
34	0.622	0.718	0.542	0.887	0.695	0.888	0.889	0.907	**0.919**
36	0.631	0.701	0.534	0.889	0.690	0.889	0.889	0.917	**0.929**
38	0.633	0.707	0.660	0.881	0.694	0.877	0.879	0.912	**0.927**
40	0.627	0.710	0.652	0.884	0.712	0.881	0.885	0.910	**0.911**
AVG	0.604	0.723	0.592	0.885	0.686	0.886	0.886	0.909	**0.921**

The bold values mean the best performance in all methods

Table 7 Precision on the PIE dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
20	0.264	0.449	0.442	0.694	0.344	0.699	0.711	0.717	**0.747**
22	0.291	0.440	0.266	0.674	0.356	0.685	0.684	0.707	**0.742**
24	0.236	0.415	0.250	0.664	0.357	0.683	0.674	0.724	**0.751**
26	0.245	0.409	0.239	0.692	0.338	0.666	0.676	**0.696**	0.690
28	0.230	0.392	0.191	0.641	0.328	0.651	0.627	0.665	**0.681**
30	0.233	0.373	0.198	0.594	0.310	0.605	0.635	0.625	**0.687**
32	0.229	0.372	0.186	0.651	0.329	0.653	0.634	0.676	**0.688**
34	0.236	0.361	0.182	0.631	0.327	0.634	0.643	0.650	**0.687**
36	0.247	0.319	0.172	0.631	0.311	0.651	0.641	0.657	**0.682**
38	0.240	0.334	0.278	0.610	0.315	0.595	0.603	0.651	**0.687**
40	0.230	0.331	0.269	0.623	0.328	0.626	0.636	0.667	**0.648**
AVG	0.244	0.381	0.243	0.646	0.331	0.650	0.651	0.676	**0.699**

The bold values mean the best performance in all methods

Table 8 Recall on the PIE dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
20	0.335	0.544	0.606	0.856	0.422	0.855	0.858	0.894	**0.911**
22	0.329	0.531	0.389	0.832	0.424	0.827	0.832	0.859	**0.898**
24	0.311	0.496	0.350	0.801	0.425	0.818	0.813	0.874	**0.883**
26	0.320	0.492	0.313	0.824	0.417	0.816	0.819	0.855	**0.894**
28	0.304	0.474	0.271	0.818	0.401	0.819	0.811	0.856	**0.881**
30	0.308	0.456	0.278	0.783	0.383	0.782	0.789	0.837	**0.874**
32	0.301	0.443	0.247	0.806	0.405	0.812	0.804	0.876	**0.876**
34	0.309	0.439	0.255	0.813	0.403	0.815	0.813	0.856	**0.885**
36	0.312	0.413	0.237	0.814	0.391	0.801	0.810	**0.872**	0.868
38	0.309	0.403	0.375	0.790	0.396	0.789	0.789	0.863	**0.867**
40	0.293	0.409	0.356	0.793	0.410	0.785	0.787	0.851	**0.870**
AVG	0.312	0.464	0.334	0.812	0.407	0.801	0.811	0.863	**0.882**

The bold values mean the best performance in all methods

Fig. 3 Some samples from the YaleB dataset contaminated by 4×4 black blocks

Table 9 AC on the YaleB dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
16	0.299	0.434	0.140	0.546	0.361	0.547	0.556	0.623	**0.672**
18	0.286	0.404	0.131	0.473	0.326	0.480	0.476	0.599	**0.6391**
20	0.307	0.404	0.128	0.522	0.336	0.514	0.529	0.621	**0.643**
22	0.279	0.380	0.125	0.526	0.339	0.520	0.521	0.609	**0.650**
24	0.274	0.371	0.121	0.480	0.328	0.469	0.475	0.587	**0.646**
26	0.268	0.341	0.113	0.478	0.316	0.485	0.479	0.587	**0.615**
28	0.268	0.340	0.113	0.455	0.320	0.463	0.452	0.590	**0.608**
30	0.255	0.323	0.108	0.464	0.319	0.460	0.468	0.583	**0.618**
32	0.261	0.321	0.105	0.469	0.304	0.480	0.478	0.591	**0.614**
34	0.243	0.320	0.105	0.469	0.317	0.471	0.468	0.595	**0.609**
36	0.246	0.311	0.104	0.488	0.326	0.493	0.487	0.588	**0.608**
38	0.230	0.306	0.106	0.471	0.300	0.463	0.496	0.585	**0.663**
AVG	0.268	0.354	0.116	0.487	0.324	0.487	0.490	0.596	**0.632**

The bold values mean the best performance in all methods

Table 10 NMI on the YaleB dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
16	0.329	0.522	0.110	0.631	0.427	0.627	0.635	0.684	**0.736**
18	0.334	0.491	0.123	0.587	0.406	0.591	0.587	0.674	**0.721**
20	0.375	0.516	0.173	0.625	0.438	0.626	0.634	0.694	**0.742**
22	0.360	0.495	0.151	0.636	0.453	0.634	0.634	0.694	**0.749**
24	0.362	0.506	0.160	0.613	0.458	0.604	0.610	0.681	**0.739**
26	0.364	0.477	0.167	0.620	0.453	0.623	0.622	0.688	**0.729**
28	0.384	0.486	0.181	0.609	0.465	0.611	0.608	0.690	**0.723**
30	0.382	0.490	0.183	0.617	0.476	0.615	0.618	0.691	**0.738**
32	0.399	0.488	0.193	0.622	0.466	0.627	0.627	0.694	**0.740**
34	0.398	0.495	0.207	0.628	0.484	0.627	0.628	0.704	**0.733**
36	0.398	0.488	0.210	0.648	0.500	0.649	0.649	0.703	**0.739**
38	0.388	0.489	0.211	0.639	0.493	0.634	0.653	0.710	**0.763**
AVG	0.373	0.495	0.172	0.623	0.460	0.622	0.625	0.692	**0.738**

The bold values mean the best performance in all methods

Table 11 Precision on the YaleB dataset contaminated by 4×4 black blocks

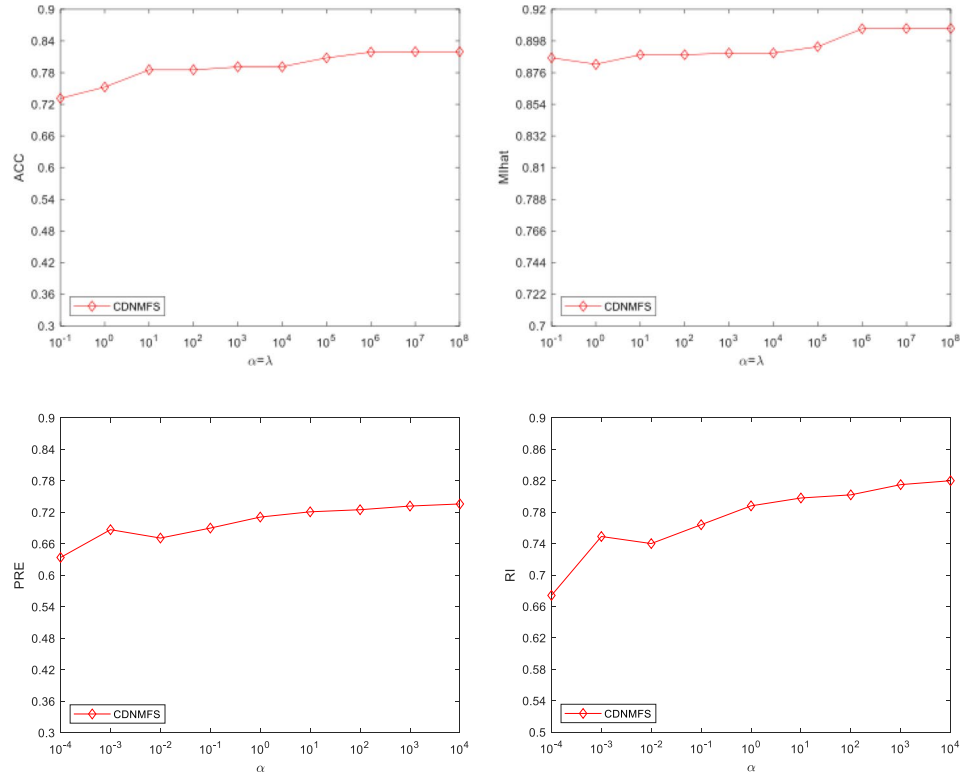
K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
16	0.161	0.296	0.269	0.410	0.223	0.406	0.416	0.469	**0.532**
18	0.152	0.266	0.249	0.333	0.199	0.342	0.337	0.423	**0.495**
20	0.165	0.268	0.248	0.373	0.207	0.373	0.387	0.439	**0.515**
22	0.139	0.240	0.224	0.362	0.207	0.356	0.363	0.437	**0.506**
24	0.134	0.235	0.193	0.331	0.196	0.321	0.331	0.412	**0.504**
26	0.124	0.202	0.164	0.335	0.183	0.335	0.337	0.408	**0.469**
28	0.127	0.199	0.177	0.298	0.183	0.297	0.293	0.399	**0.428**
30	0.121	0.189	0.161	0.311	0.180	0.311	0.311	0.398	**0.466**
32	0.124	0.184	0.147	0.312	0.167	0.323	0.321	0.387	**0.462**
34	0.114	0.184	0.132	0.311	0.173	0.312	0.311	0.402	**0.445**
36	0.114	0.175	0.140	0.331	0.183	0.333	0.331	0.391	**0.434**
38	0.097	0.162	0.135	0.305	0.166	0.298	0.342	0.412	**0.519**
AVG	0.131	0.217	0.186	0.334	0.189	0.334	0.340	0.415	**0.481**

The bold values mean the best performance in all methods

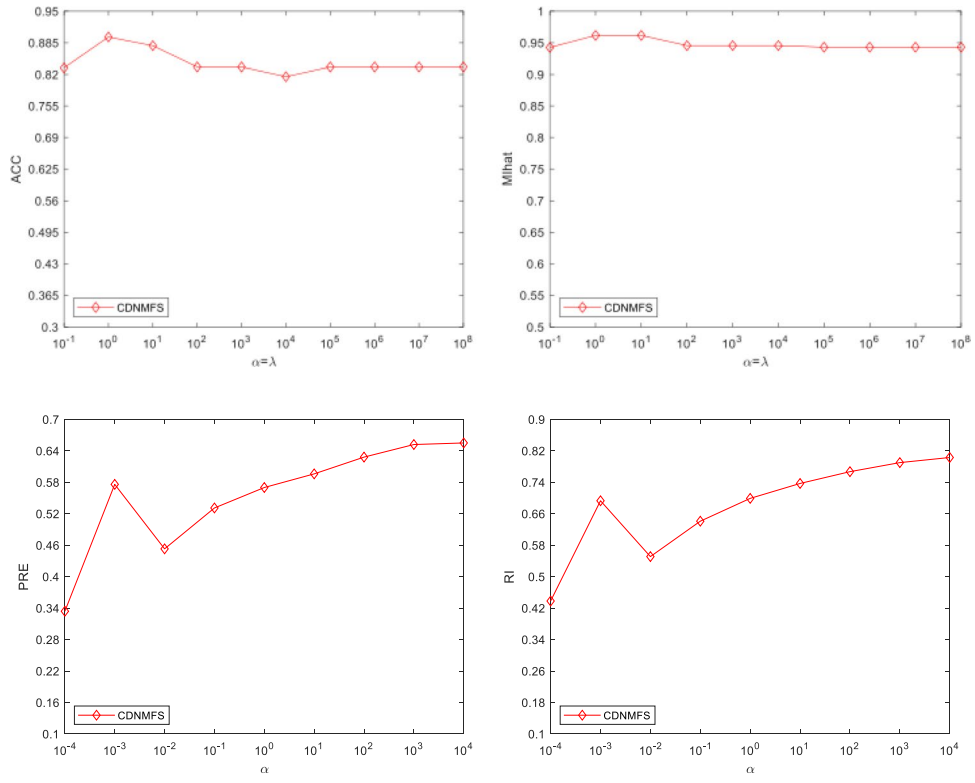
Table 12 Recall on the YaleB dataset contaminated by 4×4 black blocks

K	KM	NMF	CF	GNMF	CNMF	CGNMF	DGNMF	DGSNMF	CDNMFS
16	0.205	0.388	0.409	0.522	0.264	0.518	0.526	0.590	**0.656**
18	0.194	0.322	0.355	0.479	0.234	0.484	0.480	0.574	0.629
20	0.207	0.334	0.327	0.504	0.247	0.505	0.506	0.580	**0.637**
22	0.192	0.299	0.299	0.502	0.252	0.505	0.497	0.569	**0.653**
24	0.180	0.289	0.261	0.485	0.235	0.475	0.481	0.553	**0.611**
26	0.169	0.253	0.226	0.472	0.223	0.478	0.472	0.547	**0.591**
28	0.170	0.249	0.236	0.468	0.227	0.477	0.475	0.544	**0.602**
30	0.163	0.241	0.217	0.450	0.228	0.449	0.452	0.533	**0.599**
32	0.161	0.228	0.197	0.445	0.200	0.446	0.453	0.535	**0.592**
34	0.156	0.226	0.176	0.448	0.216	0.448	0.447	0.535	**0.582**
36	0.153	0.214	0.177	0.468	0.223	0.467	0.469	0.541	**0.591**
38	0.147	0.198	0.181	0.470	0.198	0.450	0.452	0.530	**0.575**
AVG	0.175	0.270	0.255	0.476	0.229	0.475	0.476	0.552	**0.610**

The bold values mean the best performance in all methods



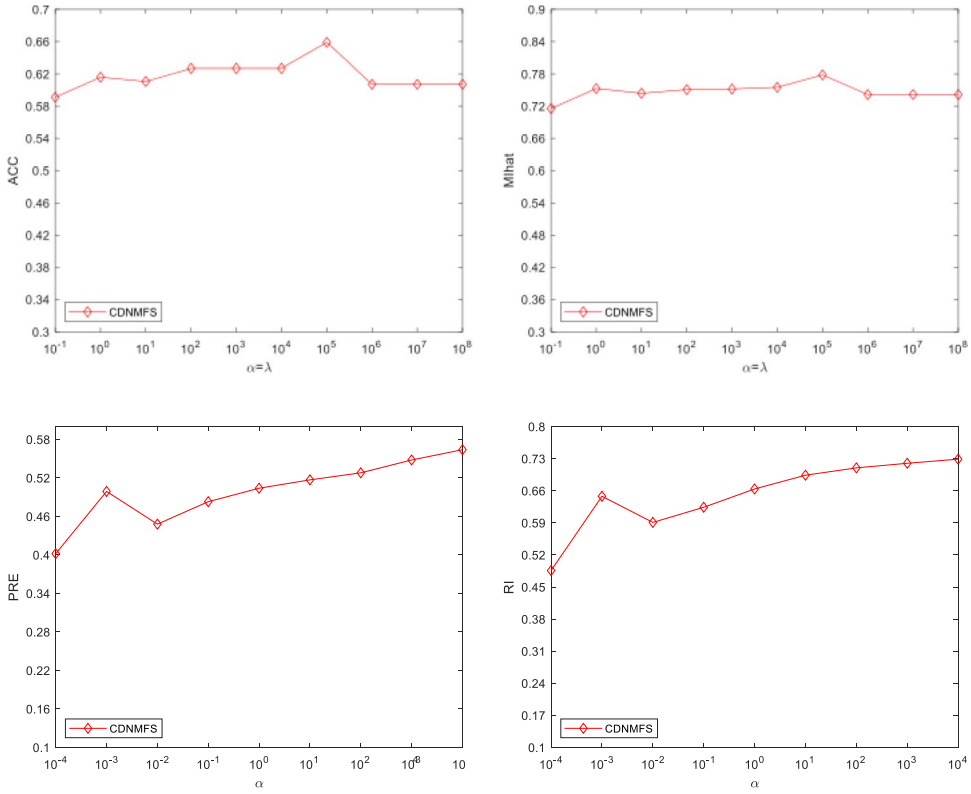
(a) COIL20



(b) PIE

Fig. 4 The performance of CDNMFS varied the parameter $\alpha = \lambda$ **a** COIL20; **b** PIE; **c** YaleB

Fig. 4 (continued)



(c) YaleB

TheYaleB dataset includes 5760 single light source images of 38 individuals under 576 viewing conditions. For every subject in a particular pose, an image with ambient (background) illumination was also captured. We simply used the cropped images and resized them to 32×32 pixels. Figure 3 shows some samples from the YaleB dataset contaminated by 4×4 black blocks.

We used experimental schemes that were similar to the two experiments above. K categorical samples were randomly chosen to carry out the experiments. Tables 9, 10, 11, and 12 show the performances of all methods under different values of the parameter K. We can see that the proposed CDNMFS method achieved the best performance among the compared algorithms under different numbers of classes. The main reason for this result is that our proposed method can effectively capture the low-dimensional structural information of the original data by considering more prior knowledge about the data.

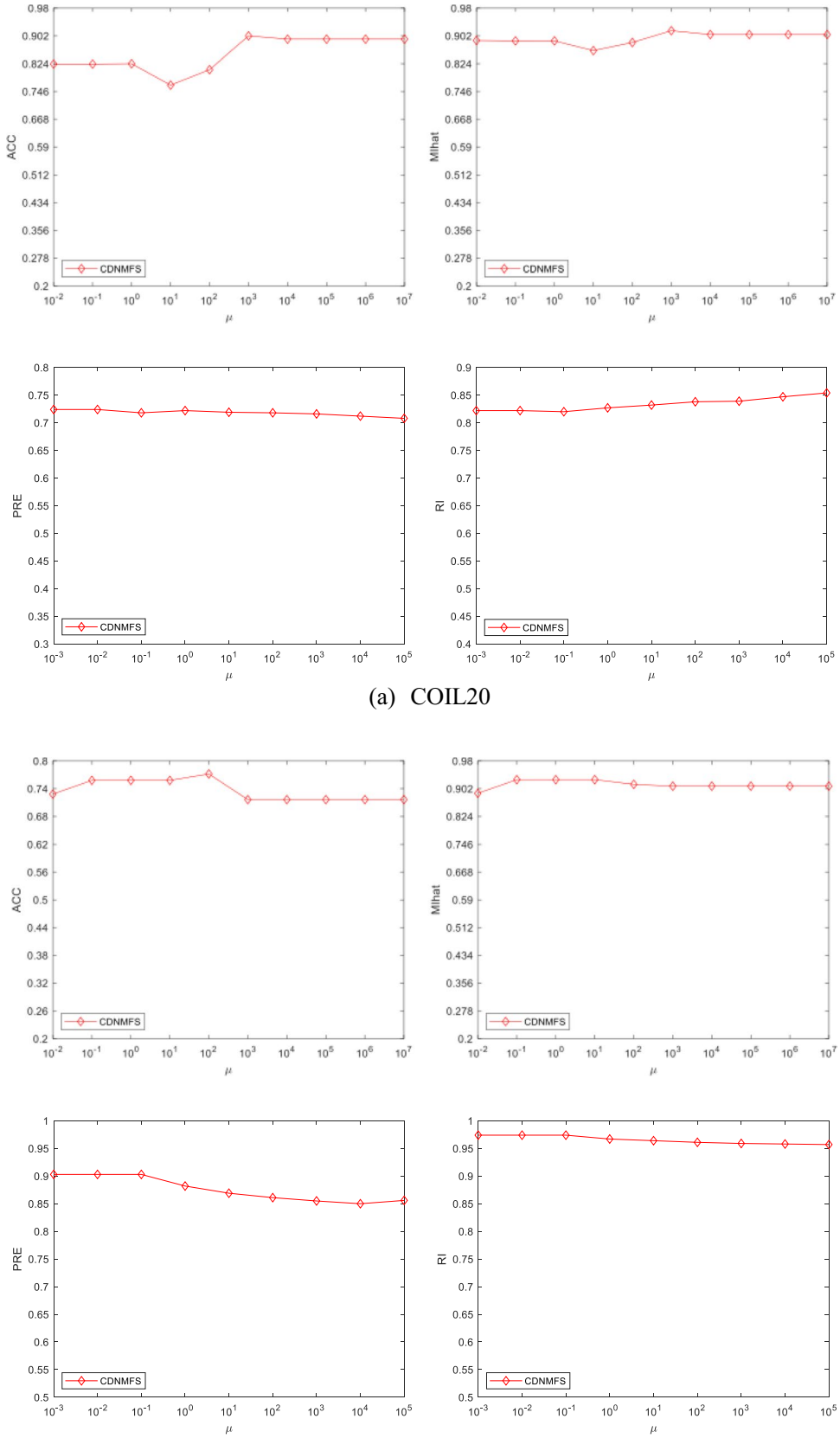
4.2 Experimental analysis

From the experimental results obtained on the three test databases, we can draw some interesting conclusions:

(1) We can observe that both GNMF and DGNMF can achieve better performance than NMF on the three real datasets. The main reason for this is that GNMF fully exploits the local manifold structure of data embedded in data by constructing a nearest neighbor graph. DGNMF takes advantage of the intrinsic manifold structure hidden in the data space as well as that in the feature space. Therefore, the experimental results demonstrate that the clustering performance can be significantly improved by fully utilizing the local geometric structure of the input data.

(2) It is clear to see that GCNMF can achieve better performance than GNMF on the three datasets, whose samples are contaminated by 4×4 black blocks. This is because GCNMF employs the correntropy instead of the Euclidean norm to measure the reconstruction error. Therefore, GCNMF is more insensitive to non-Gaussian noise and outliers in observation data than GNMF.

(3) We can see that the DGSNMF can achieve the better performance than traditional shallow learning methods in most cases on the three datasets. The main reason for this is that deep architecture can better extract the intrinsic feature information from data. Unfortunately, DGSCNMF only considers the manifold structure of

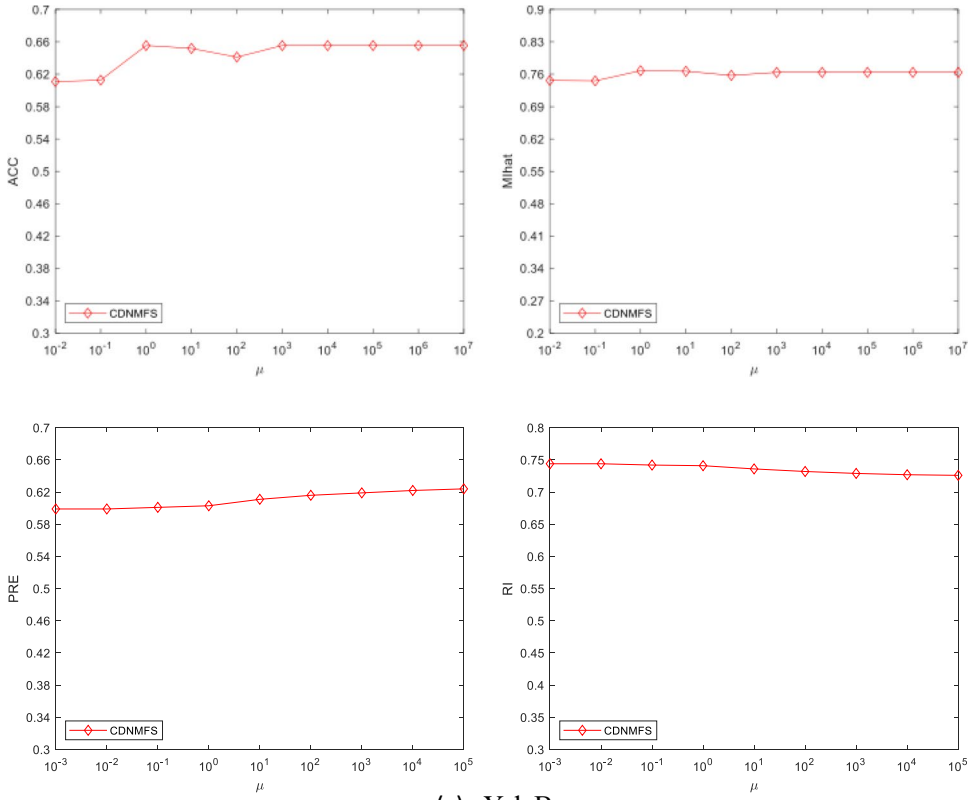


(a) COIL20

(b) PIE

Fig. 5 The performance of CDNMFS varied the parameter μ **a** COIL20; **b** PIE; **c** YaleB

Fig. 5 (continued)



(c) YaleB

data and cannot take full use of more prior knowledge hidden in the input data.

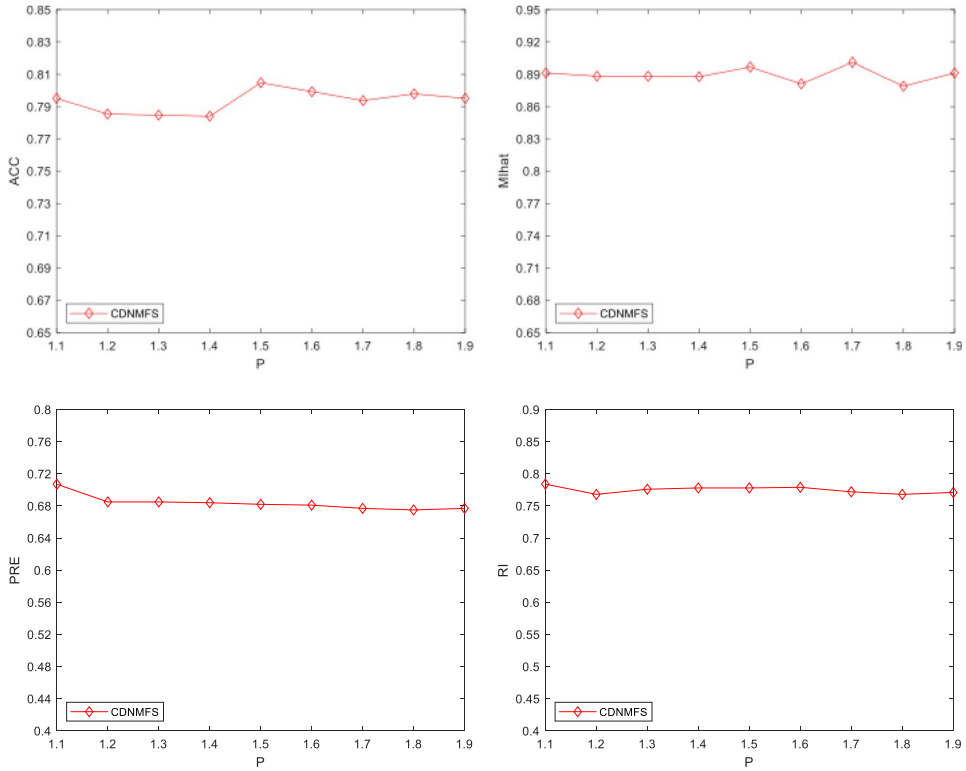
(4) From Tables 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, and 12, we can see that the proposed CDNMFS method exhibits the best average performances in the comparison with other approaches. The main reason is that CDNMFS makes full use of more prior knowledge than utilized by other methods. Specifically, the local manifold structures in the data and feature space are effectively preserved in low- dimensional representations by constructing two graph regularizers. In addition, the correntropy is also used as the measurement metric for improving the robustness of the method to non-Gaussian noise and outliers. Furthermore, the L_p norm is used to constrain the basis matrix and thus an accurate solution for the proposed model can be achieved. Therefore, our proposed CDNMFS method exhibits more representation power than other state-of-the-art competitors.

4.3 Parameters selection

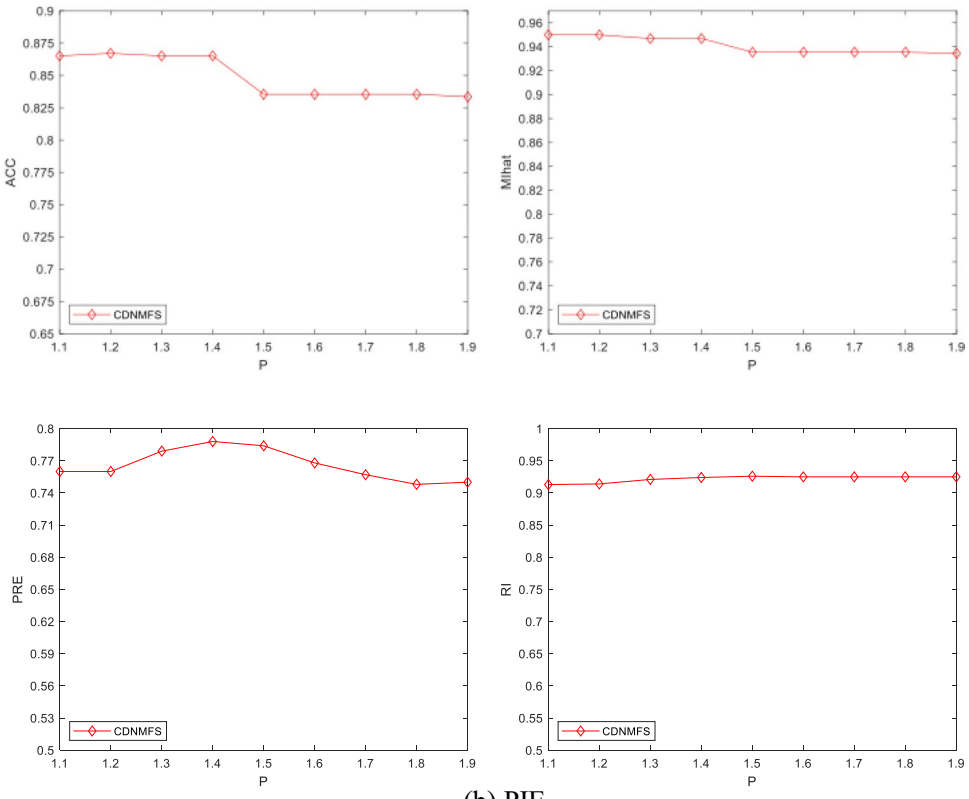
The proposed CDNMFS model contains four parameters α ,λ, μ and P. To investigate the sensitivity of each parameter clearly, we varied one parameter by fixing the other parameters. In our model, α and λ denote the regularization parameters of the dual graph, and thus we simply set their values to be equal. Figures 4, 5 and 6 show that the clustering results of the proposed CDNMFS method varied with the parameters α,λ, μ and P on the three datasets, respectively. From the results of the parameter experiments, it can be seen that CDNMFS achieved relatively stable performances varied with the parameters α, λ,μ and P.

4.4 Convergence Study

In this subsection, we experimentally study the convergence of the proposed CDNMFS method on the three aforementioned datasets. The convergence speeds of the proposed CDNMFS method for all datasets are shown in Fig. 7. In Fig. 7, the x-axis denotes the number of iterations and the y-axis denotes the objective function value. From the experimental results, it can be noticed that the CDNMFS method can almost converge after 20 iterations, which verifies the effectiveness and fast convergence of the proposed algorithm.



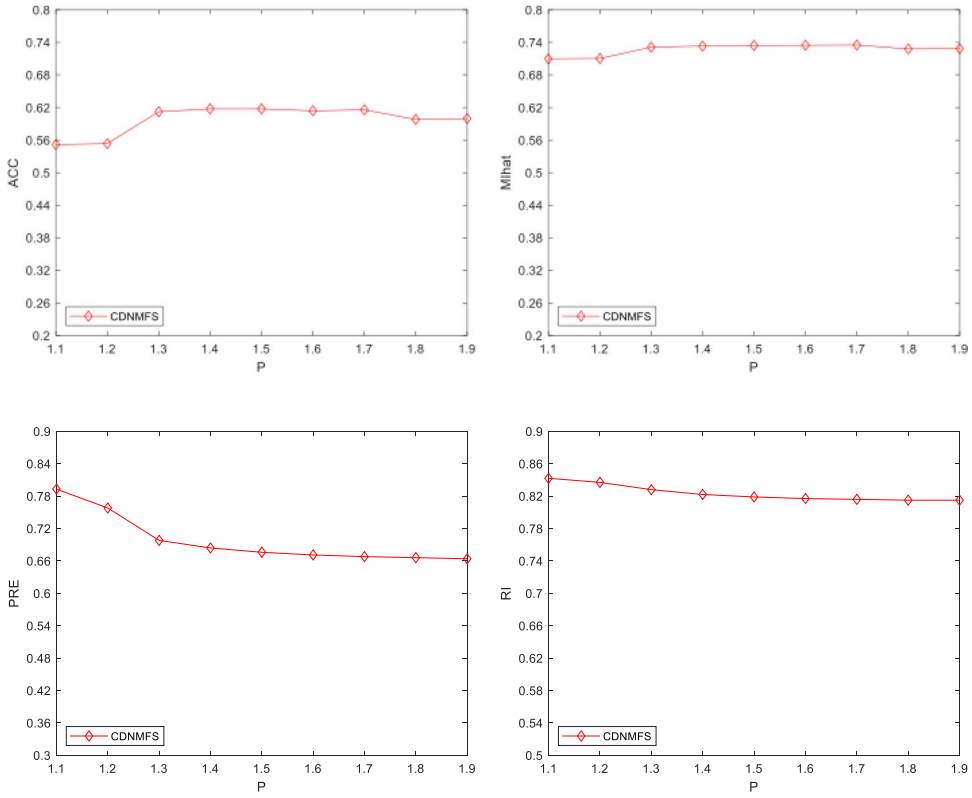
(a) COIL20



(b) PIE

Fig. 6 The performance of CDNMFS varied the parameter *P* **a** COIL20; **b** PIE; **c** YaleB

Fig. 6 (continued)



(c) YaleB

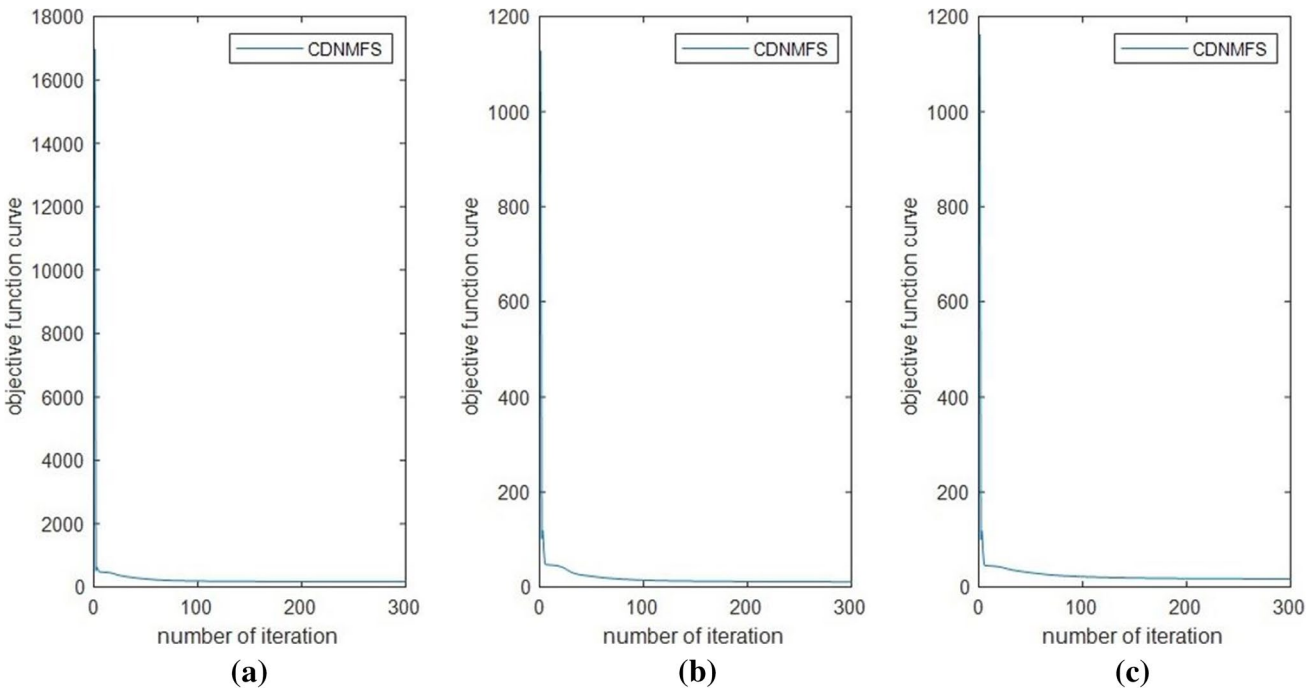


Fig. 7 Convergence speed of CDNMFS: **a** COIL20; **b** PIE; **c** YaleB

5 Conclusion

In this paper, we propose a novel algorithm, called correntropy-based dual graph regularized NMF with L_P smoothness (CDNMFS) for image representation. By adopting the correntropy as the evaluation metric, we find that the proposed CDNMFS method is robust to noise and outliers that follow a non-Gaussian distribution. In addition, it effectively discovers the intrinsic manifold structures hidden in the input data and features and imposes an L_p norm-based smoothness constraint to obtain highly accurate solutions. The experimental results on three real datasets demonstrate the effectiveness of the proposed CDNMFS method.

Acknowledgements This work was supported by the National Natural Science Foundation of China [Grant No. 61603159, 62006097, 62162033], Natural Science Foundation of Jiangsu Province [Grant No. BK20160293, BK20200593], Excellent Key Teachers of QingLan Project in Jiangsu Province.

References

1. Lei X, Tie J, Fujita H (2020) Relational completion based nonnegative matrix factorization for predicting metabolite-disease associations. Knowledge-Based Syst 204:106238.
2. Li H, Zhang SJ, Liu G (2017) Graph-based discriminative nonnegative matrix factorization with label information. Neurocomputing 266:91–100
3. Jiang B, Ding C, Luo B (2018) Robust data representation using locally linear embedding guided PCA. Neurocomputing 275:523–532
4. Deng T, Ye D, Ma R et al (2020) Low-rank local tangent space embedding for subspace clustering. Inf Sci 508:1–21
5. Xue Y, Tang Y, Xu X et al (2021) Multi-objective feature selection with missing data in classification. IEEE Trans Emerg Topic Comput Intell 10:1–10
6. Li X, Cui G, Dong Y (2017) Graph regularized non-negative low-rank matrix factorization for image clustering. IEEE Trans Cybernetics 47(11):3840–3853
7. Yang X, Jiang X, Tian C, et al (2020) Inverse projection group sparse representation for tumor classification: a low rank variation dictionary approach. Knowl Based Syst 196:105768.
8. Cai D, He X, Han J, Huang TS (2011) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 33(8):1548–1560
9. Zhang Z, Zhang Y, Liu G et al (2019) Joint label prediction based semi-supervised adaptive concept factorization for robust data representation. IEEE Trans Knowl Data Eng 32(5):952–970
10. Zhang Z, Zhang Y, Li S et al (2021) Flexible Auto-weighted Local-coordinate concept factorization: a robust framework for unsupervised clustering. IEEE Trans Knowl Data Eng 33(4):1523–1539
11. Lee DD, Seung HS (1999) Learning the parts of objects by nonnegative matrix factorization. Nature 401:788–791
12. Palmer SE (1977) Hierarchical structure in perceptual representation. Cogn Psychol 6:441–474
13. Wachsmuth E, Oram M W, Perrett DI (1994) Recognition of objects and their component parts: responses of single units in the temporal cortex of the macaque 4:509–522
14. Logothetis N, Sheinberg D (1996) Visual object recognition. Annu Rev Neurosci 19:577–621
15. Leng C, Zhang H, Cai G (2018) A novel data clustering method based on smooth non-negative matrix factorization, in ICSM, pp 24–26.
16. Chen P, He Y, Lu H, et al (2015) Constrained non-negative matrix factorization with graph Laplacian, International Conference on Neural Information Processing.
17. Guan N, Tao D, Luo Z, Yuan B (2011) Manifold regularized discriminative nonnegative matrix factorization with fast gradient descent. IEEE Trans Image Process 20:2030–2048
18. Shang F, Jiao LC, Wang F (2012) Graph dual regularization nonnegative matrix factorization for co-clustering. Pattern Recogn 45(6):2237–2250
19. He R, Zheng W, Hu B (2011) Maximum correntropy criterion for robust face recognition. IEEE Trans Pattern Anal Mach Intell 33(8):1561–1576
20. Zhou N, Dong B, et al (2019) Maximum correntropy criterion-based robust semi-supervised concept factorization for image representation. IEEE Trans Neural Networks Learn Syst.
21. Wang Y, Wu S, Mao B et al (2014) Correntropy induced metric based graph regularized non-negative matrix factorization. Neurocomputing 204(5):172–182
22. Lee DD, Seung H (2011) Algorithms for non-negative matrix factorization. Adv Neural Inform Process Syst 13.
23. Shu Z, Fan H et al (2017) Multiple Laplacian graph regularized low-rank representation with application to image representation. IET Image Proc 6:370–378
24. Chen B, Wang X, Li Y et al (2019) Maximum correntropy criterion with variable center. IEEE Signal Process Lett 99:1–1
25. Shu Z, Wu X, You C et al (2020) Rank-constrained nonnegative matrix factorization algorithm for data representation. Inf Sci 528:133–146
26. He Y, Fei W, Li Y et al (2019) Robust matrix completion via maximum correntropy criterion and Half Quadratic optimization. IEEE Trans Signal Process 68:181–195
27. Shu Z, Wu X et al (2019) The optimal graph regularized sparse coding with application to image representation. PRCV 2:91–101
28. Yang J, Cao J, Xue A (2020) Robust Maximum mixture correntropy criterion-based semi-supervised ELM with variable center. IEEE Trans Circuit Syst 99:3572–3576
29. He R, Zhen W, Hu B (2011) Maximum correntropy criterion for robust face recognition. IEEE Trans Pattern Anal Mach Intell 33(8):1561–1576
30. Trigeorgis G, Bousmalis K, Zafeiriou S et al (2017) A deep matrix factorization method for learning attribute representations[J]. IEEE Trans Pattern Anal Mach Intell 39(3):417–429