Mathematical Biosciences and Engineering

http://www.aimspress.com/journal/MBE

MBE, 19(2): 2147–2178.
DOI: 10.3934/mbe.2022100
Received: 22 September 2021
Accepted: 14 December 2021
Published: 28 December 2021

Research article

Data representation using robust nonnegative matrix factorization for edge computing

Qing Yang[1],*, Jun Chen[1] and Najla Al-Nabhan[2]

[1] School of Computer Engineering, Nanjing Institute of Technology, Hongjing Avenue, Nanjing, China
[2] Dept. Computer Science, King Saud University, Riyadh, KSA

* **Correspondence:** Email: yangq@njit.edu.cn; Tel: +86-02586118293; Fax: +8602586118293.

Abstract: As a popular data representation technique, Nonnegative matrix factorization (NMF) has been widely applied in edge computing, information retrieval and pattern recognition. Although it can learn parts-based data representations, existing NMF-based algorithms fail to integrate local and global structures of data to steer matrix factorization. Meanwhile, semi-supervised ones ignore the important role of instances from different classes in learning the representation. To solve such an issue, we propose a novel semi-supervised NMF approach via joint graph regularization and constraint propagation for edge computing, called robust constrained nonnegative matrix factorization (RCNMF), which learns robust discriminative representations by leveraging the power of both L2,1-norm NMF and constraint propagation. Specifically, RCNMF explicitly exploits global and local structures of data to make latent representations of instances involved by the same class closer and those of instances involved by different classes farther. Furthermore, RCNMF introduces the L2,1-norm cost function for addressing the problems of noise and outliers. Moreover, L2,1-norm constraints on the factorial matrix are used to ensure the new representation sparse in rows. Finally, we exploit an optimization algorithm to solve the proposed framework. The convergence of such an optimization algorithm has been proven theoretically and empirically. Empirical experiments show that the proposed RCNMF is superior to other state-of-the-art algorithms.

Keywords: nonnegative matrix factorization (NMF); edge computing; constraint propagation; L2,1-norm; clustering

1. Introduction

With the development of automatic driving technology, edge computing (EC) has attracted more and more attention. Compared with cloud computing that exploits distributed computing to decomposing many data processing tasks through the network, EC is initiated at the edge of the data source, and thus reduces the process of data transfer on the network. To recognize a face image, for example, cloud computing will upload this image to the server first, and then the cloud server will complete such a task. EC directly calculates the recognition result after obtaining the image. Obviously, it requires a variety of techniques to complete different tasks. Generally, the data EC processes usually has high-dimensional and complex structure [1]. Furthermore, in multimedia mining, pattern recognition and bio-informatics [2–4], one is often faced with high-dimensional data. Directly dealing with such high-dimensional data requires massive time and memory cost for learning tasks. In fact, the features of data are not all discriminative and important, since many of them are redundant or noisy [4–6]. Important and meaningful features always lie on (or near) a low-dimensional space [7,8]. This leads one to investigate a new technique to process these data. Dimensionality reduction (DR) is to find a low-dimensional representation of high-dimensional data by preserving the inherent desired structure contained in the original data [9–11]. It has been proved to be an effective method to decrease the dimensionality of data [12–18]. Existing popular representative methods for DR are Linear Discriminant Analysis (LDA) [19,20], Principal Component Analysis (PCA) [10], Locality Preserving Projections (LPP) [21], LLE [12], ISOMAP [13] and Laplacian Eigenmap [22], and so on. The basis vectors and coefficients obtained by these approaches above are not constrained the non-negativity and thus usually contain negative values. Data in real applications like texts, images, audios and videos, are naturally nonnegative. Basis vectors and coefficients with negative values lack of clear physical meaning and interpretability for nonnegative data [23].

As a new DR algorithm, Nonnegative Matrix Factorization (NMF) has recently been presented to solve the nonnegative data. NMF is to decompose a given nonnegative matrix into two nonnegative factor matrices, so that the product of the two factor matrices can well approximate the given one [24]. It constrains the two factors to be nonnegative. That is, all elements of two factor matrices must be greater than or equal to zero. NMF is a parts-based DR method since it only allows the addition combination of the original data space, not the subtraction combination [25]. It has successfully been used in many fields, such as face recognition [24,26], document clustering [27,28], image processing [29–31], and molecular pattern discovery [32,33]. Therefore, many improved NMF-based algorithms have been put forward. Ding et al. [34] presented Convex-NMF and Semi-NMF to increase its applicable range by relaxing the data matrix to hold positive and negative numbers, respectively. A graph regularized NMF (GNMF) [35] is put forward to promote discriminating ability of the ordinary NMF by defining an affinity graph to encode the geometrical data structure. Li et al. [36] incorporated distant repulsion and basis redundancy elimination into the cost function of GNMF and thus developed a structure preserving NMF approach. Zhang et al. [37] proposed manifold regularized low-rank matrix decomposition by extending GNMF to the nonlinear space. A structure constraint NMF [38] is developed to apply the intra-sample structures to promote the matrix decomposition process. Cichocki et al. [39] proposed a general framework for NMF based on Csizar's divergences, which improved the efficiency and robustness of NMF. Cichocki et al. [40] exploited orthogonality and sparsity constraints to extend NMF. Fevotte et al. [41] proposed a novel NMF based on the β-divergence cost function (β-NMF). Devarajan et al. [42] presented a unified method for NMF based on the generalized linear model theory. Devarajan et al. [43] used generalized dual Kullback-Leibler divergence to improve NMF and thus proposed a statistical NMF framework.

Above approaches are unsupervised and ignore supervised information, including class label and pairwise constraint. Actually, supervised information is used to promote the performance of learning methods [8,15,16,18]. Usually, class labels for all data are fairly expensive to obtain but limited supervised information is readily available, therefore semi-supervised NMF with a small amount of supervised information has attracted considerable attention [44–46]. Constrained NMF method (CNMF) [31] is a representative work of semi-supervised ones. It combined class labels of data as additional constraints to improves the discriminating power. Robust structured NMF algorithm (RSNMF) [8] applied class label to encode the block-diagonal structure, which embeds instances from the same class to the same space. constraint propagation-based semi-supervised NMF algorithm (CPSNMF) [23] propagates pairwise constraints on the entire data set in the form of cannot-link and must-link, which constructs a weight graph as the regularization to constrain the factorial matrix of NMF. Zhang et al. [47] used pairwise constraints to guide NMF clustering and thus proposed an NMF-based constrained clustering method. Yang et al. [48] developed an adaptive non-smooth NMF approach which applying using a data-related algorithm to adaptively obtain the smoothness matrix. Label propagation based semi-supervised NMF [49] is developed to integrate class label propagation and matrix decomposition into a joint model by exploring the distribution relationship between labeled and unlabeled data instances. By employing the relationship between label information and feature representation, adaptive graph semi-supervised NMF [50] enhanced the recognition ability of feature representation and completes the classification task. Li et al. [51] integrated deep learning, matrix factorization, and view fusion into a matrix framework and thus proposed a novel adaptive-weighted multi-view method. Jia et al. [52] extended the conventional NMF by exploiting the similarity and dissimilarity regularizers to steer matrix factorization. Xing et al. [53] added the label information of some data to the objective of NMF as the regularization term.

Although NMF and its variants have strong mathematical theory and encouraging performance, there are still three important problems to be further addressed. First, above-mentioned approaches fail to use global and local structures of the data to enhance the decomposition ability of NMF simultaneously, which limits their application to real-world scenes. Local and global structures are applied to find a discriminative and compact representation [15,54]. CNMF and RSNMF ignored the local geometrical structure of data when they expanded NMF to semi-supervised scenario. Generally, the local geometric structure of data is are supposed to be locally invariant in data set. That is, nearest neighbor samples should have the same class label [33,55]. Both GNMF and RMNMF [45] only pay close attention to the local structure and ignore the global structure, thus losing discriminating power. Second, most semi-supervised NMF methods neglect the important role of instances from different classes in learning the representation of data. Like instances with the same class label, instances with different class labels can also provide discriminative information. In fact, it makes the result more intuitive and interpretable that instances involved in the same class are mapped closer, while those involved in different classes are mapped farther in the new representation space. Third, above-mentioned approaches become unstable and have disappointed performances when processed data contain noise and outliers. Because the residual of every instance gets into the loss function of NMF in the form of square, it is easy to generate outliers [56,57]. Kong et al. [58] put forward robust NMF to get around this problem, which applies the $L_{2,1}$-norm cost function to achieve nonnegative matrix decomposition. Generally, real data in many applications are contaminated by noise and outliers.

Therefore, it is necessary and important to investigate a novel algorithm to improve the robustness of NMF.

To address the above problems, we put forward a novel NMF algorithm for data representation, called robust constrained nonnegative matrix factorization (RCNMF), which discover the intrinsic geometric and discriminating structure of data in semi-supervised scenario. In the proposed RCNMF model, pairwise constraints are propagated to unlabeled instances for encoding global and local structures. Then, we construct a specific intrinsic graph to depict the between-cluster separability and the within-cluster compactness in the latent representation space. Additionally, we introduce the L2,1-norm cost function to enhance the robustness of the new model. Hence, RCNMF can be naturally applied to practical learning tasks. The main contributions of the proposed algorithm are worth highlighting here:

1) Our algorithm explicitly exploits global and local structures of the original data space via constraint propagation and merges it into our model to guide matrix decomposition. The proposed RCNMF approach makes latent representations of instances involved by the same class closer and those of instances involved by different classes farther. Thus, the proposed RCNMF has more discriminating ability than other semi-supervised ones.

2) Different from CNMF, NMFCC and CPSNMF which are not robust to outliers and noise, the new model can address the problem of noise and outliers since RCNMF exploit the $L_{2,1}$-norm loss to achieve the nonnegative matrix decomposition. Moreover, the new model imposes the $L_{2,1}$-norm on the coding matrix to ensure the new representation sparse in rows.

3) The proposed RCNMF can take advantage of the ability of $L_{2,1}$-norm NMF and constraint propagation, and characterize it as an optimization problem. The new model explores an efficient iterative algorithm to solve such an optimization problem with the theoretical analyses and the experimental verification. In addition, it is easy to observe that several algorithms, such as RNMF and CPSNMF, are special cases of the proposed algorithm. Thus, our method is a general framework.

4) Our method propagates cannot-link and must-link constraints to unlabeled samples, thus can get more supervised information. Moreover, the new model can adapt to class label and pairwise constraint scenarios. This naturally leads to various applications.

The rest of this paper is structured as follows. We briefly review related work in Section 2. We elaborate the proposed RCNMF model and the corresponding optimization in Section 3. We provide convergence proof and computational complexity of RCNMF in Section 4. Extensive experiments on clustering are reported in Section 5. Finally, we give some conclusions in Section 6.

2. Related work

Given an original nonnegative data matrix $X = [x_1, \ldots, x_N] \in \mathbb{R}^{M\times N}$ with N instances, $x_i \in \mathbb{R}^M$ is an instance vector. NMF is to find two nonnegative matrix factors $U = [u_{ik}] \in \mathbb{R}^{M\times K}$ and $V = [v_{jk}] \in \mathbb{R}^{N\times K}$ so that

$$X \approx UV^T \tag{1}$$

NMF makes use of the square of the Euclidean distance to solve the optimal approximation U and V, respectively. The Euclidean square distance of NMF is defined as [24, 59]:

$$\min_{U\geq 0, V\geq 0} ||X - UV^T||_F^2 \tag{2}$$

where $||\cdot||_F$ is the Frobenius norm of the matrix.

The problem (2) is not convex for optimizing two factors *U* and *V* simultaneously, but is convex when one variable is fixed and the other is solved. Thus, Lee and Seung [24] presented two iterative update rules to optimize the problem (2), which is formulated as:

$$u_{ik} \leftarrow u_{ik} \frac{(XV)_{ik}}{(UV^T V)_{ik}} \tag{3}$$

$$v_{jk} \leftarrow v_{jk} \frac{(X^T U)_{jk}}{(VU^T U)_{jk}} \tag{4}$$

2.1. Manifold nonnegative matrix factorization (*MNMF*)

Cai et al. proposed the graph regularized NMF (GNMF) to GNMF [35] explicitly takes into account the local invariance of data and sets up the nearest neighbor graph to formulate the geometrical structure of the original data space. It depicts this graph structure as the Laplacian matrix, which is added to the loss function of NMF as the regularization term. Thus, GNMF optimizes the following problem:

$$\min_{U\geq 0, V\geq 0} ||X - UV^T||_F^2 + \lambda Tr(V^T L V) \tag{5}$$

where$\lambda \geq 0$ is a regularization parameter and $Tr(\cdot)$ denotes the matrix trace. $L = D - W$ is a Laplacian matrix of the neighbor graph. *W* denotes a connection weight matrix of the nearby instances and *D* denotes a diagonal matrix whose entries $d_{ii} = \sum_j w_{ij}$. Two alternate rules are exploited to iteratively solve the optimal model (5). The updating formulas are respectively described as followed:

$$u_{ik} \leftarrow u_{ik} \frac{(XV)_{ik}}{(UV^T V)_{ik}} \tag{6}$$

$$v_{jk} \leftarrow v_{jk} \frac{(X^T U + \lambda WV)_{jk}}{(VU^T U + \lambda DV)_{jk}} \tag{7}$$

Zhang et al. [37] expended GNMF to the nonlinear space for characterizing the nonlinear structure of the data, which is stated as followed:

$$\min_{U^T U = I, V} ||X - UV^T||_F^2 + \lambda Tr(V^T \Phi V) \tag{8}$$

where Φ can be a Laplacian matrix, e.g., formulated as the weight matrix $I - W$ in the LLE [12].

Huang et al. [56] integrated GNMF and $L_{2,1}$-NMF [47] into a joint framework and proposed a robust manifold NMF method (RMNMF). Wu et al. [59] also presented a robust manifold NMF algorithm similar to RMNMF. The objective function in [59] increases the constraint $U \geq 0$ and reduces the orthogonal constraint in comparison to RMNMF. We call the above four methods as Manifold Nonnegative Matrix Factorization (MNMF), since they also construct the nearest neighbor graph to depict the local information of the input space.

2.2. NMF-based constrained clustering (NMFCC)

Zhang et al. [47] put forward an NMF-based constrained clustering method, called NMFCC. NMFCC enforces the similarity between two instances belonging to a must-link constraint to approach one and that of two instances belonging to a cannot-link constraint to approach zero, which is composed of a similarity matrix. It uses the square of the class indicator matrix to compute the similarity matrix and regards as a regularization term to cluster. NMFCC minimizes the following problem:

$$\underset{U\geq 0,V\geq 0}{J(U,V)} = \|X - UV^T\|^2 + \|A \circ (VV^T - Q)\|^2 \tag{9}$$

where $\circ$ refers to the dot product between two matrixes and Q denotes the constraint matrix whose entry is 1 if $(i,j) \in ML$ (must-link set) or $i = j$ else 0. A is a coefficient matrix defined as

$$A_{ij} = \begin{cases} \alpha, (i,j) \in ML \: or \: i \\ \beta, (i,j) \in CL \\ 0, otherwise \end{cases} \tag{10}$$

After the Lagrange function for Eq (10) is constructed and used to solve the derivatives with respect to *U* and *V*, NMFCC get the update rules of Eqs (11) and (12):

$$u_{ik} \leftarrow u_{ik} \frac{(XV)_{ik}}{(UV^TV)_{ik}} \tag{11}$$

$$v_{jk} \leftarrow v_{jk} \frac{\Upsilon_{jk}}{\Pi_{jk}} \tag{12}$$

where

$$\Upsilon = X^TU + 2(A \circ Q \circ A^T)V + 4(A \circ A^T)(V \circ V \circ V)$$

$$\Pi = VU^TU + 2(A \circ (VV^T) \circ A^T)V + 4(A \circ A^T)(V \circ V \circ V) \tag{13}$$

2.3. Constrained nonnegative matrix factorization (CNMF)

Liu et al. [31] put forward a new semi-supervised NMF method, called CNMF. Different from the above-mentioned methods in this section, CNMF treats a small amount of class label as hard constraints to ensure that instances with the same label are embedded into the same low-dimensional space. Given *l* labeled instances and *n-l* unlabeled instances, CNMF first constructs the indicator matrix *C* where $c_{ij} = 1$ if x_i belongs to class *j*; $c_{ij} = 0$ otherwise. The matrix ***A*** of label constraints is formulated as

$$A = \begin{pmatrix} C_{l\times c} & 0 \\ 0 & I_{n-l} \end{pmatrix}$$

CNMF introduces an auxiliary matrix ***Z*** to incorporate label constraint information and redefines a coefficient matrix ***V*** as *V* = *AZ*. Hence, CNMF solves the following problem:

$$\min_{U\geq 0,V\geq 0} \|X - UZ^TA^T\|_F^2 \tag{14}$$

The solutions of CNMF are expressed as:

$$u_{ik} \leftarrow u_{ik} \frac{(XAZ)_{ik}}{(UZ^T A^T AZ)_{ik}} \tag{15}$$

$$z_{jk} \leftarrow z_{jk} \frac{(A^T X^T U)_{jk}}{(A^T AZU^T U)_{jk}} \tag{16}$$

2.4. Discriminative nonnegative matrix factorization (*DNMF*)

Babaee et al. [60] presented a label constrained NMF method, called Discriminative Nonnegative Matrix Factorization (DNMF). Similar to CNMF, DNMF first constructs the indicator matrix $Q \in \mathbb{R}^{S\times N}$where $q_{ij} = 1$ if x_i belongs to class j; $q_{ij} = 0$ otherwise. Different from CNMF, unlabeled instances are assigned 0 in Q. Then, DNMF defines the label information constraint as:

$$\|Q - AV_l^T\|_F^2$$

where $V_l = [v_1, \dots, v_c, 0, \dots, 0]^T \in \mathbb{R}^{N\times K}$and the matrix **A** which is calculated as part of the NMF optimization. Therefore, the formulation of DNMF is as follows:

$$\min_{U\geq 0, V\geq 0} \|X - UV^T\|_F^2 + \lambda\|Q - AV_l^T\|_F^2 \tag{17}$$

DNMF uses the following iterative update rules and obtains the corresponding *U*, *V*, and *A*:

$$u_{ik} \leftarrow u_{ik} \frac{(XV)_{ik}}{(UV^T V)_{ik}} \tag{18}$$

$$v_{jk} \leftarrow v_{jk} \frac{(X^T U + \lambda(V_l A^T A)^- + \lambda(Q^T A)^+)_{jk}}{(VU^T U + \lambda(V_l A^T A)^+ + \lambda(Q^T A)^-)_{jk}} \tag{19}$$

$$A \leftarrow QV_l(V_l^T V_l)^{-1} \tag{20}$$

2.5. Robust structured nonnegative matrix factorization (*RSNMF*)

A semi-supervised robust structured NMF algorithm (RSNMF) [8] is proposed to arrive at the separated low-dimensional space. The key idea of RSNMF is to use the block-diagonal structure of the labeled instances to increase the discriminating ability. Specifically, RSNMF introduces an indicator matrix $I = [\bar{I}, \hat{0}] \in \mathbb{R}^{r\times n}$, where $\hat{0} \in \mathbb{R}^{r\times u}$ is a zero matrix for the unlabeled instances. Given labeled instances, the definition of $\bar{I} \in \mathbb{R}^{r\times l}$ is expressed as follows:

$$\bar{\mathrm{I}} = \begin{pmatrix} \bar{0}_1 & 1 & \cdots & 1 \\ 1 & \bar{0}_2 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & \bar{0}_c \end{pmatrix}$$

in which $\bar{0}_c \in \mathbb{R}^{m\times n_c}$ is the zero-matrix corresponding to *c*-th class. RSNMF describes the block-diagonal structure as $\|I \odot V\|_F^2$, where $\odot$ is the elementwise multiplication operator. RSNMF defines the following loss function:

$$\min_{U\geq 0,V\geq 0}\frac{1}{2}\|X-UV^T\|_{2,p}^p+\frac{\lambda}{2}\|I\odot V\|_F^2 \tag{21}$$

Thus, RSNMF designs the following updating rules for these two matrix factors:

$$u_{ik}\leftarrow u_{ik}\frac{(XDV)_{ik}}{(UV^TDV)_{ik}} \tag{22}$$

$$v_{jk}\leftarrow v_{jk}\frac{(DX^TU)_{jk}}{(DVU^TU)_{jk}+\lambda(I\odot V)_{jk}} \tag{23}$$

where $d_{kk}=p/2\|z_k\|_2^{2-p}$ and $Z=X-UV^T$.

3. Robust constrained NMF

As mentioned above, NMF and its variants are unsupervised algorithms, and fail to exploit some supervised information to guide the decomposition progress and to improve the discriminating ability. In fact, utilizing a small amount of supervised information has always been an important issue in many fields of computer vision and machine learning [15,16,18]. Generally, supervised information has various forms. The two commonly used forms are pairwise constraint and class label, respectively. In this paper, we exploit cannot-link and must-link constraints to conduct matrix decomposition. A must-link constraint specifies that two instances have the same cluster label. A cannot-link constraint indicates that two instances have different cluster labels. In many applications, it is more practical to get pairwise constraints than to get class labels, because users can easily demonstrate whether two samples belong to the same cluster [16,31]. Besides, one can use the class label of the sample to obtain pairwise constraint, but not vice versa. Consequently, pairwise constraints are weaker supervised information than class label.

Inspired by recent research on NMF and semi-supervise learning [23,61], in this paper, we propose a novel robust constrained nonnegative matrix factorization (RCNMF), which explicitly combines constraint propagation and L2,1-norm NMF in a new way. We use cannot-link and must-link constraints to propagate over the whole data set, thus obtain the constraint information as a regularizer added to the L2,1-norm NMF. Next, we describe the RCNMF model in detail.

3.1. $L_{2,1}$ norm NMF

For any matrix A, its j-th column, i-th row are denoted by a_j and a^i, respectively. Tr (B) means the trace of B if B is square. A^T is the transposed matrix of A. I denotes an identity matrix and 1 is a column vector whose entries are all 1. We define the $L_{2,1}$ norm of the matrix B as [62]:

$$\|B\|_{2,1}=\sum_{i=1}^n\sqrt{\sum_{j=1}^m B_{ji}^2}=\sum_{i=1}^n\|b_i\|_2 \tag{24}$$

One can use NMF and the $L_{2,1}$ norm to group the data matrix X into k clusters $(C_1,\dots C_k)$ as:

$$\min_{U,V}\|X-UV^T\|_{2,1}$$

$$s.t. \mathrm{U}\geq 0, V\geq 0, V\in\{0,1\}^{n\times k}, V^T1=1 \tag{25}$$

where *U* denotes the basis matrix and *V* is the encoding matrix. Because of the constraints on *V*, the optimization problem (25) is difficult to solve. A commonly used method is to relax this constraint to orthogonality, i.e., $V^TV=I$. Thus, the problem (25) is rewritten as:

$$\min_{U,V} \|X - UV^T\|_{2,1}$$

$$s.t. \mathrm{U} \geq 0, V \geq 0, V^TV = I \tag{26}$$

3.2. Constraint propagation

For the data set $X = [x_1, \ldots, x_N] \in \mathbb{R}^{M \times N}$, we describe initial must-link constraints as $ML = \{(x_i, x_j)|l_i = l_j\}$ and initial cannot-link constraints as $CL = \{(x_i, x_j)|l_i \neq l_j\}$, where l_i denote the clustering label of the instance x_i. To propagate both cannot-link and must-link constraints on the whole data set, Lu et al. [61] used "+1" and "-1" to represent the difference between the two types of constraints. The propagation operation aims to determine the constraint weight between two unconstrained instances, which essentially clusters instances between classes marked with +1 or classes marked with -1. Thus, the initial pairwise constraint matrix $Z = [z_{ij}] \in \mathbb{R}^{N \times N}$ is denoted as:

$$z_{ij} = \begin{cases} +1, (x_i, x_j) \in ML \\ -1, (x_i, x_j) \in CL \\ 0, \text{otherwise} \end{cases} \tag{27}$$

The matrix of propagated pairwise constraints is defined as $F = [f_{ij}] \in \mathbb{R}^{N \times N}$, where $|f_{ij}| \leq 1$. The matrix *F* represents a set of pairwise constraints with relevant clustering weight.$f_{ij} \geq 0$ means $(x_i, x_j) \in ML$, while $f_{ij} \leq 0$ means $(x_i, x_j) \in CL$. Constraint propagation in [61] is formulated as:

$$\min_F \|F - Z\|_F^2 + \frac{\mu}{2} Tr(F^TLF + FLF^T) \tag{28}$$

The specific algorithm of constraint propagation is formulated as:

1) Set up a nearest neighbor graph vis denoting its nearby weight $W \in \mathbb{R}^{n \times n}$ as:

$$w_{ij} = \frac{a(x_i, x_j)}{\sqrt{a(x_i, x_i)}\sqrt{a(x_j, x_j)}}$$

if $x_i \in N_p(x_j)$ or $x_j \in N_p(x_i)$ and $w_{ij} = 0$, otherwise, where $a(x_i, x_j) = \exp(-\|x_i - x_j\|^2/\sigma)$ and $N_p(x_i)$ denotes the set of p nearest neighbors of x_i.

2) Compute the matrix $L = D^{-1/2}WD^{-1/2}$, where D is a diagonal matrix whose *i*-th diagonal entry is $\sum_j w_{ij}$.

3) Compute the vertical constraint matrix $F_v(t+1) = \eta LF_v(t) + (1-\eta)Z$ until convergence, in which α is limited to (0,1).

4) Iterate the horizontal constraint matrix $F_h(t+1) = \eta F_h(t)L + (1-\eta)F_v^*$ until convergence[1], in which F_v^* denotes the limit of $\{F_v(t)\}$.

5) $F^* = F_h^*$ is the final representation of F, where F_h^* denotes the limit of $\{F_h(t)\}$.

6) Compute the new weight matrix $\overline{W} = [\overline{w}_{ij}] \in \mathbb{R}^{n \times n}$ by utilizing F^* and the original normalized

[1] Please refer to [61] for the proof of the convergence.

weight matrix W:

$$\bar{w}_{ij} = \begin{cases} 1-(1-f_{ij}^*)(1-w_{ij}) & f_{ij}^* \geq 0 \\ (1+f_{ij}^*)w_{ij} & f_{ij}^* < 0 \end{cases} \tag{29}$$

According to the above analysis, the learned weight matrix has several distinct advantages:
1) The matrix $\bar{W}$ is symmetric and nonnegative, and $\bar{w}_{ij} \in [0,1]$. Thus, it can well describe the relationship between instances.
2) $\bar{w}_{ij} \geq w_{ij}$ (or $< w_{ij}$) if $f_{ij}^* \geq 0$ (< 0). This shows that $\bar{W}$ is derived from W. It is increased when $f_{ij}^* \geq 0$, and decreased when $f_{ij}^* < 0$.
3) $\frac{\partial \bar{w}_{ij}}{\partial w_{ij}} = 1 - |f_{ij}^*|$. This indicates that cannot-link and must-link constraints an equally significant role in computing the relationship between instances.

By using the constraint propagation approach, local and global structures is taken into account so that more pairwise constraint is obtained. Clearly, the propagated pairwise constraint matrix $\bar{W}$ characterizes that nearby instance have the same clustering label, and instances with the same structure have the same clustering label. Then, one can use pairwise constraint information to build a new weight matrix in which instances involved by the same class have larger weight values and those involved by different classes have smaller weight values.

3.3. Robust constrained nonnegative matrix factorization

We apply constraint propagation to guide the matrix decomposition process in the $L_{2,1}$-norm NMF. For dimensionality reduction or classification, there are two kinds of consistency. One is that latent representations are considered locally consistent. That is, nearby representations are supposed to have the same label. Specifically, if the two representations v_i and v_j are close to each other, they belong to the same class. A k-nearest neighbor graph is constructed to figure the similarity between nearby instances. The other is that latent representations should be globally consistent, i.e., instances with the same structure have the same label. We introduce constraint propagation to find an appropriate propagated pairwise constraint matrix $\boldsymbol{F}$ for meeting the two properties. In summary, we construct a new weight graph to model the relationship between latent representations with the weight matrix $\bar{W}$:

$$J(V) = \frac{1}{2}\sum_{i,j=1}^{n} \left\| v_i - v_j \right\|^2 \bar{w}_{ij} \tag{30}$$

If the two instances x_i and x_j have the same label, they should be close to each other in the input data space. f_{ij}^* has a large positive value so that $\bar{w}_{ij}$ also has a relatively large value in the light of Eq (29). Minimizing $J(V)$, the distance between the latent representations v_i and v_j should be small. Thus, v_i and v_j are neighbor in the low-dimensional space. On the other hand, if the two instances x_i and x_j belong to different classes, they should be kept away in the original data space. f_{ij}^* has a large negative value and thus $\bar{w}_{ij}$ has a relatively small value, which indicates that v_i and v_j are far from each other. Therefore, minimization Eq (30) makes the distance between data points belonging to must-link constraint as small as possible and the distance between data points belonging to must-link constraint as large as possible. Besides, the problem (30) has the same formulation as spectral dimensionality reduction [21,22], which plays a significant role in semi-supervised manifold learning algorithms and spectral clustering.

Combining Eqs (25) and (30), we propose a general NMF framework, called robust constrained nonnegative matrix factorization (RCNMF). RCNMF optimizes the following problem:

$$\min_{U,V}\|X-UV^T\|_{2,1}+\frac{\alpha}{4}\sum_{i,j=1}^{n}\left\|v_i-v_j\right\|^2\bar{w}_{ij}+\beta\|U\|_{2,1}$$

$$s.t.\ U\geq 0,\ V\geq 0\ ,\ V^TV=I \tag{31}$$

where α and β are two trade-off parameters. We use the $L_{2,1}$ norm on the basis matrix U as the regularization term to select the latent representation, since U is sparse and easy to be controlled by noise features.

3.4. Optimization algorithm

Like other NMF-based methods, the objective function in the model (31) is not convex for optimizing two factors and simultaneously, but is convex when one variable is fixed and the other is solved. To this end, we introduce an iterative optimization to solve optimizing these two factors. We rewrite the objective function of RCNMF as follows:

$$O=\|X-UV^T\|_{2,1}+\frac{\alpha}{4}\sum_{i,j=1}^{n}\left\|v_i-v_j\right\|^2\bar{w}_{ij}+\beta\|U\|_{2,1}$$

$$=\|X-UV^T\|_{2,1}+\frac{\alpha}{2}\sum_{i=1}^{n}v_i^Tv_i\bar{d}_{ii}-\frac{\alpha}{2}\sum_{i,j=1}^{n}v_i^Tv_j\bar{w}_{ij}+\beta\|U\|_{2,1}$$

$$=\|X-UV^T\|_{2,1}+\frac{\alpha}{2}Tr(V^T\bar{D}V)\ -\frac{\alpha}{2}Tr(V^T\bar{W}V)\ +\beta\|U\|_{2,1}$$

$$=\|X-UV^T\|_{2,1}+\frac{\alpha}{2}Tr(V^T\bar{L}V)\ +\beta\|U\|_{2,1} \tag{32}$$

where $\bar{D}$ is a diagonal matrix with$\bar{d}_{ii}=\sum_j\bar{w}_{ij}$, and $\bar{L}=\bar{D}-\bar{W}$. ψ_{ik} and ϕ_{jk} are used as the Lagrangian multiplier of two matrix factors U and V, respectively, where $\Psi=[\psi_{ik}]$ and $\Phi=[\phi_{jk}]$. $\lambda>0$ is a parameter to adjust the weight of the orthogonality condition. The corresponding Lagrange function $\mathcal{L}$ is written as

$$\mathcal{L}=\|X-UV^T\|_{2,1}+\frac{\alpha}{2}Tr(V^T\bar{L}V)+\beta\|U\|_{2,1}$$

$$+\frac{\lambda}{4}\|V^TV-I\|_F^2+Tr(\Psi U^T)+Tr(\Phi V^T) \tag{33}$$

By using $\|A\|_F^2=Tr(AA^T)$, the partial derivative of $\mathcal{L}$ with respect to U and V is calculated as

$$\frac{\partial\mathcal{L}}{\partial U}=-XEV+UV^TEV+\beta HU+\Psi \tag{34}$$

$$\frac{\partial\mathcal{L}}{\partial V}=-EX^TU+EVU^TU+\alpha\bar{L}V+\lambda V(V^TV-I)+\Phi \tag{35}$$

where *E* and *H* are diagonal matrices and their diagonal elements are given by

$$e_{ii} = 1/\|x_i - U(V^T)_i\|_2 \tag{36}$$

$$h_{ii} = 1/\|u_i\|_2 \tag{37}$$

Applying the Karush-Kuhn-Tucker (KKT) conditions $\varphi_{ik}u_{ik} = 0$ and $\phi_{jk}v_{jk} = 0$, we obtain two equations for u_{ik} and v_{jk} as follows:

$$-(XEV)_{ik}u_{ik} + (UV^TEV)_{ik}u_{ik} + \beta(HU)_{ik}u_{ik} = 0 \tag{38}$$

$$-(EX^TU)_{jk}v_{jk} + (EVU^TU)_{jk}v_{jk} + \alpha(\bar{L}V)_{jk}v_{jk} + \lambda(VV^TV - V)_{jk}v_{jk} = 0 \tag{39}$$

Thus, we get the following updating rules from Eqs (38) and (39):

$$u_{ik} \leftarrow u_{ik}\frac{(XEV)_{ik}}{(UV^TEV+\beta HU)_{ik}} \tag{40}$$

$$v_{jk} \leftarrow v_{jk}\frac{(EX^TU+\alpha\bar{W}V+\lambda V)_{jk}}{(EVU^TU+\alpha\bar{D}V+\lambda VV^TV)_{jk}} \tag{41}$$

Based on the above analysis, we can update U and V iteratively with Eqs (40) and (41) until the objective value of Eq (31) remains unchanged. The detailed optimization algorithm is described in Algorithm 1.

```
Input: Data matrix $X \in \mathbb{R}^{M\times N}$; cannot-link constraints CL and must-link constraints ML,
       Parameters $\alpha$, $\beta$, $\lambda$, $p$, and $1 \le K \le \min(M, N)$.
Output: $U \in \mathbb{R}^{M\times K}$, $V \in \mathbb{R}^{N\times K}$.
1. Initialize $U^0$, $V^0$.
2. Calculate the weight matrix $\bar{W}$ by using the constraint propagation method.
3. repeat
    3.1 Update U with
        $u_{ik}^{t+1} = u_{ik}^t \frac{(XE^tV^t)_{ik}}{(U^t(V^t)^TE^tV^t+\beta H^tU^t)_{ik}}$;
    3.2 Update V with
        $v_{jk}^{t+1} = v_{jk}^t \frac{(E^tX^TU^t+\alpha\bar{W}V^t+\lambda V^t)_{jk}}{(E^tV^t(U^t)^TU^t+\alpha\bar{D}V^t+\lambda V^t(V^t)^TV^t)_{jk}}$;
    3.3 Update the diagonal matrix E as
        $E^{t+1} = \begin{bmatrix} e_{11}^t & & \\ & \cdots & \\ & & e_{nn}^t \end{bmatrix}$, which $e_{ii}^t = 1/\|x_i - U^t((V^t)^T)_i\|_2$;
    3.4 Update the diagonal matrix H as
        $H^{t+1} = \begin{bmatrix} 1/\|u_1^t\|_2 & & \\ & \cdots & \\ & & 1/\|u_m^t\|_2 \end{bmatrix}$;
    3.5 t = t + 1;
4. until stopping criteria of U and V is satisfied.
```

We can combine the KL divergence cost function with structure learning and the regularizer of the two factors. A novel KL divergence cost function is defined as

$$\begin{aligned} D_{KL}(X\|UV^T) &= \sum_{i,j}(x_{ij}\log\frac{x_{ij}}{\sum_k u_{ik}v_{jk}} - x_{ij} + \sum_k u_{ik}v_{jk}) \\ &\quad + \frac{\alpha}{2}\sum_{i,j=1}^{N}\sum_{k=1}^{K}(v_{ik}\log\frac{v_{ik}}{v_{jk}} + v_{jk}\log\frac{v_{jk}}{v_{ik}})\,\bar{w}_{ij} \\ &\quad + \beta\sqrt{\sum_{i,k}u_{ik}^2} + \lambda\sum_{i,j}z_{ij} \end{aligned} \tag{42}$$

where $Z = [z_{ij}] = V^TV$.

Assuming that ψ_{ik} and ϕ_{jk} are the Lagrange multipliers for $u_{ik} > 0$ and $v_{jk} > 0$, we can define the following Lagrange functions:

$$\begin{aligned} \mathcal{L} &= \sum_{ij}(x_{ij}\log\frac{x_{ij}}{\sum_k u_{ik}v_{jk}} - x_{ij} + \sum_k u_{ik}v_{jk}) \\ &\quad + \frac{\alpha}{2}\sum_{i,j=1}^{N}\sum_{k=1}^{K}(v_{ik}\log\frac{v_{ik}}{v_{jk}} + v_{jk}\log\frac{v_{jk}}{v_{ik}})\,\bar{w}_{ij} \\ &\quad + \beta\sqrt{\sum_{i,k}u_{ik}^2} + \lambda\sum_{i,j}z_{ij} + \psi_{ik}u_{ik} + \phi_{jk}v_{jk} \end{aligned} \tag{43}$$

The partial derivatives of Eq (43) with respect to u_{ik} and v_{jk} are

$$\frac{\partial\mathcal{L}}{\partial u_{ik}} = \sum_l v_{lk} - \sum_l\frac{x_{il}v_{lk}}{\sum_j u_{ij}v_{lj}} + \beta\sum_p\frac{u_{ip}}{\sum_q u_{qp}} + \psi_{ik} \tag{44}$$

$$\frac{\partial\mathcal{L}}{\partial v_{jk}} = \sum_l u_{lk} - \sum_l\frac{x_{lj}u_{lk}}{\sum_i u_{li}v_{ji}} + \frac{\alpha}{2}\sum_i\left(\log\frac{v_{jk}}{v_{ik}} + 1 - \frac{v_{ik}}{v_{jk}}\right)\bar{w}_{ij} + 2\lambda\sum_i v_{ik} + \phi_{jk} \tag{45}$$

If two instances x_i and x_j are adjacent, their corresponding low-dimensional representations v_i and v_j are close to each other. Thus, v_{ik}/v_{jk} is close to 1. We adopt the following approximation:

$$\log(a) \approx 1 - \frac{1}{a}, a \to 1 \tag{46}$$

With Eq (46), we can rewrite Eq (45) as

$$\frac{\partial\mathcal{L}}{\partial v_{jk}} = \sum_l u_{lk} - \sum_l\frac{x_{lj}u_{lk}}{\sum_i u_{li}v_{ji}} + \frac{\alpha}{v_{jk}}\sum_i(v_{jk} - v_{ik})\bar{w}_{ij} + 2\lambda\sum_i v_{ik} + \phi_{jk} \tag{47}$$

Since $\bar{L} = \bar{D} - \bar{W}$, we can easily verify that $\sum_i(v_{jk} - v_{ik})\bar{w}_{ij}$ is equal to the j-th element of vector $\bar{L}V_k$. Eq (47) is formulated as

$$\frac{\partial\mathcal{L}}{\partial v_{jk}} = \sum_l u_{lk} - \sum_l\frac{x_{lj}u_{lk}}{\sum_i u_{li}v_{ji}} + \frac{\alpha}{v_{jk}}\sum_i(\,\bar{L}V_k)_j + 2\lambda\sum_i v_{ik} + \phi_{jk} \tag{48}$$

We can obtain the following equations for u_{ik} and v_{jk} by adopting the KKT conditions $\psi_{ik}u_{ik} = 0$ and $\phi_{jk}v_{jk} = 0$:

$$(\sum_l v_{lk})u_{ik} - (\sum_l\frac{x_{il}v_{lk}}{\sum_j u_{ij}v_{lj}})u_{ik} + \beta(\sum_p\frac{u_{ip}}{\sum_q u_{qp}})u_{ik} = 0 \tag{49}$$

$$(\sum_l u_{lk})v_{jk} - (\sum_l \frac{x_{il}v_{lk}}{\sum_i u_{ij}v_{lj}})v_{jk} + \alpha(\bar{L}V_k)_j + 2\lambda(\sum_i v_{ik})v_{jk} = 0 \tag{50}$$

Two updating rules can be obtained:

$$u_{ik} \leftarrow u_{ik}\frac{\sum_l(x_{il}v_{lk}/\sum_j u_{ij}v_{lj})}{\sum_l v_{lk}+\beta\sum_p(u_{ip}/\sum_q u_{qp})} \tag{51}$$

$$V_k \leftarrow (\sum_l u_{lk}\, I + 2\lambda\sum_i v_{ik}\, I + \alpha\bar{L})^{-1}\begin{bmatrix} v_{1k}\sum_l(x_{l1}u_{lk}/\sum_i u_{li}v_{1i}) \\ \vdots \\ v_{Nk}\ \sum_l(x_{lN}u_{lk}/\sum_i u_{li}v_{Ni}) \end{bmatrix} \tag{52}$$

We call the proposed algorithm based on the KL divergence cost function as RCNMF-KL.

4. Convergence and analysis

4.1. Proof of convergence

To get the optimal solution, we theoretically analyze the convergence of the proposed RCNMF. Similar to other NMF-based methods, we afford the convergence proof of the monotone property of the new model (31) under the alternate update rules of Eqs (40) and (41).

Theorem 1. The cost function O of Eq (31) is non-increasing under the alternate updating rules of Eqs (40) and (41).

To prove Theorem 1, we give a lemma about the auxiliary function.

Lemma 1. If $G(x,x')$ is an auxiliary function for $F(x)$ and meets the conditions $G(x,x') \geq F(x)$ and $G(x,x) = F(x)$, then $F(x)$ is non-increasing under the following update:

$$x^{t+1} = \underset{x}{\operatorname{argmin}} G(x,x^t) \tag{53}$$

Proof.

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

We first demonstrate that the cost function O of Eq (31) is decreasing under the update rule Eq (41) while fixing U with an appropriate auxiliary function. The cost function of RCNMF in Eq (31) can be rewritten

$$O = \|X - UV^T\|_{2,1} + \frac{\alpha}{2}Tr(V^T\bar{L}V) + \beta\|U\|_{2,1}$$

$$= Tr((X - UV^T)E(X - UV^T)^T) \ + \frac{\alpha}{2}Tr(V^T\bar{L}V) + \beta Tr(U^THU)$$

$$= Tr(XEX^T - 2UV^TEX^T) + Tr(V^TEVU^TU)$$

$$+ \frac{\alpha}{2}Tr(V^T\bar{L}V) \ + \beta Tr(U^THU) \tag{54}$$

where $e_{ii} = 1/\|x_i - U(V^T)_i\|_2$ and $h_{ii} = 1/\|u_i\|_2$.

With the help of (54), the auxiliary function regarding V is described as the following Lemma 2.

Lemma 2. Function

$$G(V,V') = Tr(XEX^T - 2UV^TEX^T) + \sum_{j=1}^{n}\sum_{k=1}^{K}\frac{(EV'U^TU)_{jk}V_{jk}^2}{V'_{jk}}$$

$$+\frac{\alpha}{2}Tr(V^T\bar{L}V) + \beta Tr(U^THU) \tag{55}$$

is an auxiliary function for $F(V)$ which

$$F(V) = Tr(XEX^T - 2UV^TEX^T) + Tr(V^TEVU^TU)$$

$$+\frac{\alpha}{2}Tr(V^T\bar{L}V) + \beta Tr(U^THU) \tag{56}$$

Proof. Clearly, $G(V,V) = F(V)$. We need to prove $G(V,V') \geq F(V)$ in the light of the definition of the auxiliary function. With Eqs (55) and (56), we find that $G(V,V') \geq F(V)$ is equivalent to

$$\sum_{j=1}^{n}\sum_{k=1}^{K}\frac{(EV'U^TU)_{jk}V_{jk}^2}{V'_{jk}} \geq Tr(V^TEVU^TU) \tag{57}$$

We exploit the following matrix inequality from the Lemma in [34]

$$\sum_{j=1}^{n}\sum_{k=1}^{K}\frac{(AS'B)_{jk}S_{jk}^2}{S'_{jk}} \geq Tr(S^TASB) \tag{58}$$

where $A \in \mathbb{R}_+^{N\times N}$, $B \in \mathbb{R}_+^{K\times K}$, $S \in \mathbb{R}_+^{N\times K}$, $S' \in \mathbb{R}_+^{N\times K}$, and $A = A^T$, $B = B^T$. Ding et al. [34] proved that Eq (58) holds. The equality holds when $S = S'$.

Setting $A = E$, $B = U^TU$, $S = V$, $S' = V'$ in Eq (58), then Eq (57) holds and $G(V,V') \geq F(V)$. Clearly, $G(V,V')$ in Eq (55) is an auxiliary function for $F(V)$ in Eq (56). Thus, we have

$$F(V^{t+1}) \leq F(V^t) \tag{59}$$

This competes the proof of Lemma 2.

Next, we prove that the cost function O of (31) is decreasing under the updating rule (40) while fixing *V*. For the convenience of description, we use U_t to represent the solution of *t*-th iteration.

Lemma 3. Let U_{t+1} be the (t+1)-th value of *U* (on the left-hand-size of (40)) and U_t be the t-th value of *U* (on the right-hand-size of (40)). With the updating rule (40), inequality (60) holds

$$Tr((X - U_{t+1}V^T)E^t(X - U_{t+1}V^T)^T) + \frac{\alpha}{2}Tr(V^T\bar{L}V) + \beta Tr(U_{t+1}^TH^tU_{t+1})$$

$$\leq Tr((X - U_tV^T)E^t(X - U_tV^T)^T) + \frac{\alpha}{2}Tr(V^T\bar{L}V) + \beta Tr(U_t^TH^tU_t) \tag{60}$$

where $e_{ii} = 1/\|x^i - u^iV^T\|_2$ and $h_{ii} = 1/\|u^i\|_2$.

Proof. It is easy to verify that Eq (40) is the solution to the following model:

$$\min_{U\geq 0} Tr((X - UV^T)E(X - UV^T)^T) + \frac{\alpha}{2}Tr(V^T\bar{L}V) + \beta Tr(U^THU) \tag{61}$$

Thus, in the *t*-th iteration, when *V* is fixed, we get

$$U^{t+1} = \underset{U\geq 0}{\operatorname{argmin}}\, Tr((X - U_tV_t^T)E_t(X - U_tV_t^T)^T)$$

$$+\frac{\alpha}{2}Tr(V_t^T\bar{L}V_t) + \beta Tr(U_t^T H_t U_t) \tag{62}$$

which is equivalent to

$$Tr((X - U_{t+1}V_t^T)E_t(X - U_{t+1}V_t^T)^T) \; + \beta Tr(U_{t+1}^T H_t U_{t+1})$$

$$\leq Tr((X - U_tV_t^T)E_t(X - U_tV_t^T)^T) \; + \beta Tr(U_t^T H_t U_t) \tag{63}$$

That is to say,

$$\sum_i \frac{\|x^i - u_{t+1}^i V_t^T\|_2^2}{2\|x^i - u_t^i V_t^T\|_2} + \beta \sum_i \frac{\|u_{t+1}^i\|_2^2}{2\|u_t^i\|_2} \leq \sum_i \frac{\|x^i - u_t^i V_t^T\|_2^2}{2\|x^i - u_t^i V_t^T\|_2} + \beta \sum_i \frac{\|u_t^i\|_2^2}{2\|u_t^i\|_2}$$

$$\Rightarrow \|X - U_{t+1}V_t^T\|_{2,1} - (\|X - U_{t+1}V_t^T\|_{2,1} - \sum_i \frac{\|x^i - u_{t+1}^i V_t^T\|_2^2}{2\|x^i - u_t^i V_t^T\|_2})$$

$$+\beta\|U_{t+1}\|_{2,1} - \beta(\|U_{t+1}\|_{2,1} - \sum_i \frac{\|u_{t+1}^i\|_2^2}{2\|u_t^i\|_2})$$

$$\leq \|X - U_tV_t^T\|_{2,1} - (\|X - U_tV_t^T\|_{2,1} - \sum_i \frac{\|x^i - u_t^i V_t^T\|_2^2}{2\|x^i - u_t^i V_t^T\|_2})$$

$$+\beta\|U_t\|_{2,1} - \beta(\|U_t\|_{2,1} - \sum_i \frac{\|u_t^i\|_2^2}{2\|u_t^i\|_2}) \tag{64}$$

According to the Lemmas in [6], $\sqrt{a} - \frac{a}{2\sqrt{b}} \leq \sqrt{b} - \frac{b}{2\sqrt{b}}$, we arrive at

$$\|X - U_{t+1}V_t^T\|_{2,1} - \sum_i \frac{\|x^i - u_{t+1}^i V_t^T\|_2^2}{2\|x^i - u_t^i V_t^T\|_2} \leq \|X - U_tV_t^T\|_{2,1} - \sum_i \frac{\|x^i - u_t^i V_t^T\|_2^2}{2\|x^i - u_t^i V_t^T\|_2} \tag{65}$$

and

$$\|U_{t+1}\|_{2,1} - \sum_i \frac{\|u_{t+1}^i\|_2^2}{2\|u_t^i\|_2} \;\leq\; \|U_t\|_{2,1} - \sum_i \frac{\|u_t^i\|_2^2}{2\|u_t^i\|_2} \tag{66}$$

Combining Eqs (62) to (64), we obtain

$$\|X - U_{t+1}V_t^T\|_{2,1} + \frac{\alpha}{2}Tr(V_t^T\bar{L}V_t) + \beta\|U_{t+1}\|_{2,1}$$

$$\leq \|X - U_tV_t^T\|_{2,1} + \frac{\alpha}{2}Tr(V_t^T\bar{L}V_t) + +\beta\|U_t\|_{2,1} \tag{67}$$

This completes the proof of Lemma 3. Thus, we arrive at

$$F(U^{t+1}) \leq F(U^{t}) \tag{68}$$

Based on Eqs (59) and (68), Theorem 1 is proved. Thus, we can obtain that the cost function in Eq (31) is non-increasing via exploiting the update rules of Eqs (40) and (41).

4.2. Complexity analysis

In this section, the computational complexity of the proposed method will be analyzed. Generally, a capital symbol O is used to formulate the complexity of one approach.

According to the update rules of Eqs (40) and (41), the complexities to compute U and V are O $(MNK+MK)$ and O $(MNK+NK)$, respectively. If the proposed algorithm converges after t operations, the total computational complexity of calculating U and V is O $(tNMK)$. After t iterations, the computational costs of E and H are O $(t(NM+N))$ and O $(t(MK+N))$ respectively, since they are diagonal matrices. Besides, the RCNMF method also costs O (MN^2) to construct the k-nearest neighbor graph and O (pN^2) to propagate constraints. Thus, the total computational complexity of the RCNMF method is O $(tMNK + t\,(NM+N) + t\,(MK+M) + (M + p)\,N^2)$.

5. Experiments

In this section, we present the experimental results and analysis of the proposed RCNMF. Following [8, 23,31,35], we verify the performance of our new method on the basis of clustering.

5.1. Experimental setting

Table 1. Description of the data set.

Data sets	Instances	Features	Classes
UMIST	575	644	20
YaleB	2,414	1,024	38
ORL	400	644	40
USPS	9,298	256	10
MNIST	4,560	784	10
Isolet	1,560	617	26

In our experiments, we apply six publicly available data sets to compare the performance of the RCNMF and other state-of-the-art methods. These data sets are derived from different application scenarios, including three face image data sets, i.e., UMIST, YaleB and ORL, two digital image data sets, i.e., USPS and MNIST, one Letter image data set Isolet. Negative elements in Isolet and USPS are set to zeros. There are 400 grayscale images of 40 objects in the ORL data set. we resize those images to 28 × 23. Thus, the feature size of each image is 644. The statistics of the six data sets are shown in Table 1.

To evaluate the superiority of our RCNMF for data representation, we compare it with six representative NMF-based methods, including unsupervised and semi-supervised ones. The compared methods are described below.

1) GNMF [35]: the GNMF encodes the local geometrical information of the data to the NMF objective function in an unsupervised learning scenario and is regarded as the baseline for comparison here.
2) MNMFL$_{21}$ [59]: It utilizes the L2,1 norm to measure the quality of matrix factorization and the geometrical structure of the data to consider the local invariance; it is the sparse version of NMF.
3) CNMF [31]: It takes the class label as additional hard constraints to combine instances with the same class label in the low-dimensional space so that the part-based representation obtained by this method has the same label as the input data.
4) NMFCC [47]: It constructs a cost function to punish the violation of cannot-link and must-link constraints, which is considered as a regularization term added to NMF.
5) CPSNMF [23]: It applies pairwise constraint to preserve the geometrical structure of the input space, which promote the performance of the original NMF; it actually extends the GNMF to semi-supervised scenarios via using constraint propagation.
6) RSNMF [8]: It uses labeled instances to set up block-diagonal structure for learning the image representation so that instances involved in the same class are projected in the same low-dimensional subspace.

5.2. Evaluation metric

After data presentations are learned, we comprehensively verify the performance of several methods in terms of two popular metrics widely applied in both DR and clustering approaches, i.e., normalized mutual information (NMI) and accuracy (ACC). For an approach, the higher NMI and ACC are, the better the clustering performance is. Given two random variables Y and Z, NMI [8,23,63] is defined as

$$NMI = \frac{I(Y,Z)}{\sqrt{H(Y)H(Z)}} \tag{69}$$

where Y and Z represent the weight matrix of the real data label and clustering indicator provided by the algorithm, respectively. Specifically, if clustering indicator is achieved, NMI can be rewritten as

$$NMI = \frac{\sum_{i=1}^{K}\sum_{j=1}^{K} n_{ij}\log(\frac{n \cdot n_{ij}}{n_i \cdot \hat{n}_j})}{\sqrt{(\sum_{i=1}^{K} n_i \log\frac{n_i}{n})(\sum_{j=1}^{K} \hat{n}_j \log(\frac{\hat{n}_j}{n}))}} \tag{70}$$

where n_i denotes the number of instances belonging to C_i $(1 \le i \le K)$ cluster, $\hat{n}_j$ is the number of instances involved in the j-th class $(1 \le j \le K)$, and n_{ij} describes the number of instances that are located at the intersection between the j-th class and cluster C_i.

The other metric is ACC that is employed to test the percentage of the result obtained by the clustering method. If the real data label and the clustering indicator are expressed as s_i and r_i respectively, ACC [8,23] is formulated as

$$ACC = \frac{1}{n}\sum_{i=1}^{n} \delta(s_i, map(r_i)) \tag{71}$$

where $\delta(a,b) = 1$ if $a = b$ and $\delta(a,b) = 0$ otherwise. $map(r_i)$ is a permutation mapping function, in which the clustering indicator r_i is mapped into the real data label s_i.

5.3. Parameter description

CNMF does not require any parameters. MNMFL21 and GNMF have the graph regularization parameter α and the nearest neighbor size k. CPSNMF has a propagation parameter η except the above two parameters. NMFCC has two regularization parameters α and β. RCNMF has two regularization parameters, besides k and η. For CPSNMF, MNMFL21, RCNMF, and GNMF, we specify the neighborhood size by setting $k = 5$ for all the data sets. Following CPSNMF, the propagation parameter η is set to 0.2. We choose the graph regularization parameter α within $\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$ for all the compared algorithms. For RCNMF, to satisfy the orthogonality, we set λ = 105 in the experiment. RSNMF has three essential parameters, the regularization parameter α, the norm parameter q and the dimensionality parameter m. According to RSNMF, the parameter q and m are set to 0.5 and 2, respectively. Values of various parameters are shown in Table 2.

Table 2. Parameter selection in various algorithms.

Parameters	α	k	β	λ	q	m
GNMF	$\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$	5	-	-	-	-
MNMFL21	$\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$	5	-	-	-	-
CNMF	-	-	-	-	-	-
NMFCC	1	-	1	-	-	-
CPSNMF	$\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$	5	-	-	-	-
RSNMF	-	-	-	-	[0.5, 2]	[0.5, 2]
RCNMF	$\{10^{-3}, 10^{-2}, 10^{-1}, 1, 10^{1}, 10^{2}, 10^{3}\}$	5	[0.1, 100]	10^{5}	-	-

5.4. Results and analysis

To obtain the comprehensive comparison, we will conduct two types of experiments. One is that the evaluation is performed with different cluster numbers. The other is that the evaluation is done with different pairwise constraints. We describe the experiments as follow.

1) Since those typical NMF methods exploit lots of supervised information to guide matrix factorization, for the fair comparison, we apply ground-truth label to generate pairwise constraints to seek these two matrix factors. Specifically, we can randomly select 10% of all the given instances as labeled ones, which meets the requirements of CNMF and RSNMF. These labeled data are employed to generate pairwise constraints for NMFCC, CPSNMF and our RCNMF.

2) For the first type of experiment, we randomly choose 10% labeled instances for every category in the data set except the ORL data set. For ORL, we choose 20% labeled instances following [8,31], because each class has only one image if 10% labeled instances are chosen. The chosen labeled instances are applied to generate the corresponding cannot-link and must-link constraints. We randomly select different clusters k ($1 \le k \le K$) from each data set to check the performance of seven compared algorithms. For the second type of experiment, we take the number of ground-truth class as cluster numbers k for every data set. Furthermore, we randomly choose different labeled instances for

each category in the data set to check the effectiveness of the compared algorithms.

3) For fair comparison, we exploit a random strategy to initialize these two matrix factors U and V. Following [8,23,31,47,59], we use the classical K-means algorithm to cluster the learned coefficient matrix in the low-dimensional data space.

4) For given cluster number k, labeled instances, and pairwise constraints, we perform 20 experiments independently and report the average clustering results of these 20 experiments.

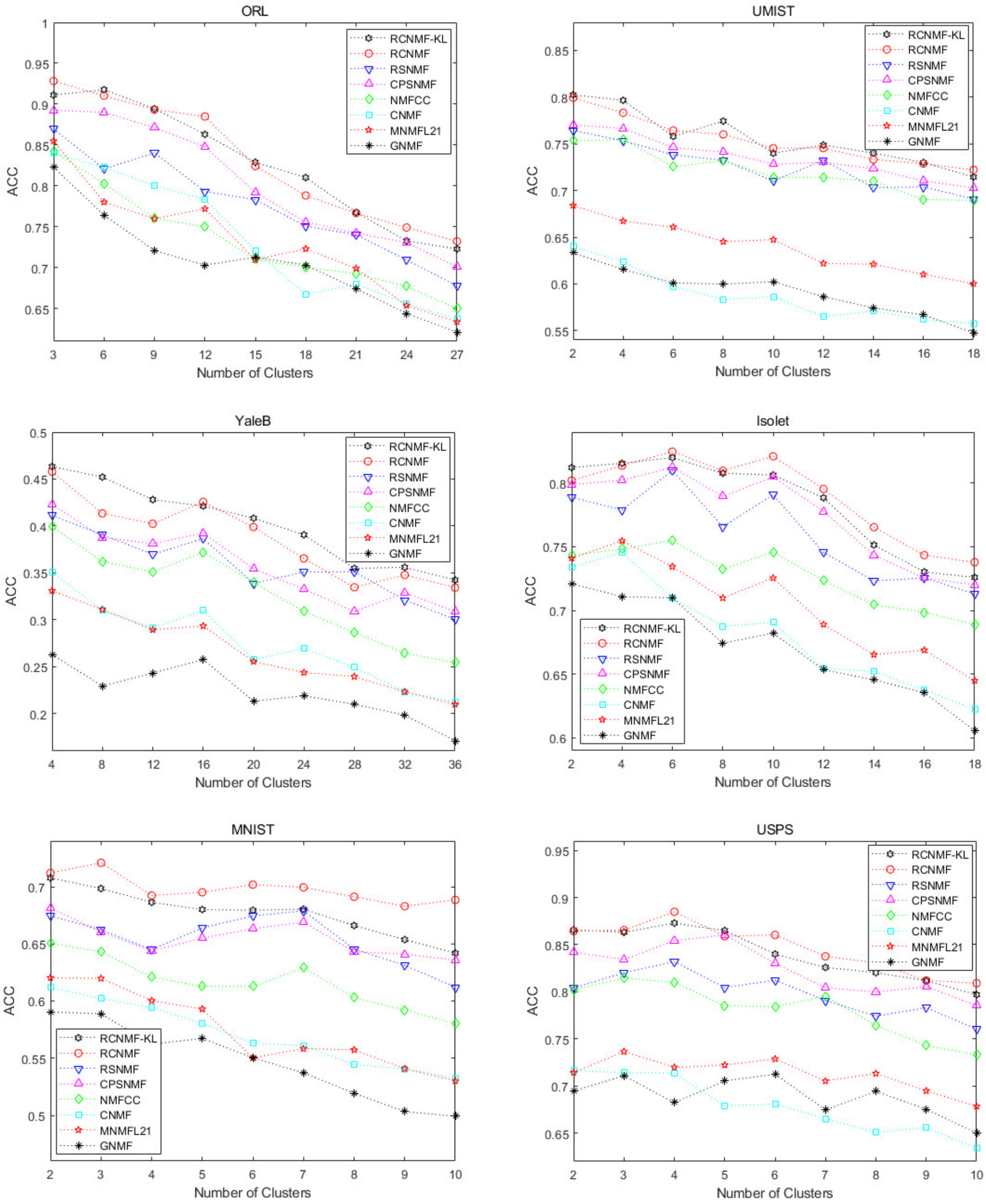

Figure 1. Clustering results by the ACC measurement with different numbers of clusters.

We conducted all experiments on an Intel CPU E5-1650 DUAL 3.60GHz with 16.0 GB RAM. Computational times for all algorithms are shown in Table 3. Figures 1 and 2 illustrate the graphical clustering results of ACC and NMI with different cluster numbers on the UMIST, YaleB, ORL, USPS, MNIST, and Isolet data sets, respectively. Figures 3 and 4 show the clustering results of ACC and NMI with different numbers of labeled instances on the UMIST, YaleB, USPS, Isolet, and MNIST data sets, respectively. In the second type of experiment, we do not compare the clustering performance of seven algorithms on ORL in the light of different numbers of labeled instances, since there are only 10 instances in each category of ORL. From the experimental results, a few interesting observations can be observed.

Table 3. Computational times (s) for all algorithms on six data sets.

Data sets	UMIST	YaleB	ORL	USPS	MNIST	Isolet
GNMF	0.0578	0.6877	0.0528	2.7606	1.6742	0.1197
MNMFL$_{21}$	1.4451	20.1605	1.4242	85.8574	59.2882	5.2622
CNMF	1.1249	14.3571	1.0710	35.6431	23.3347	1.5924
NMFCC	1.2513	15.7325	1.1744	58.508	39.0053	2.553
CPSNMF	1.3225	16.6679	1.2164	60.8361	41.9885	2.7354
RSNMF	1.5513	20.766	1.5157	87.3309	60.1338	5.5434
RCNMF	1.7196	22.1852	1.7086	92.5287	63.7294	6.8898

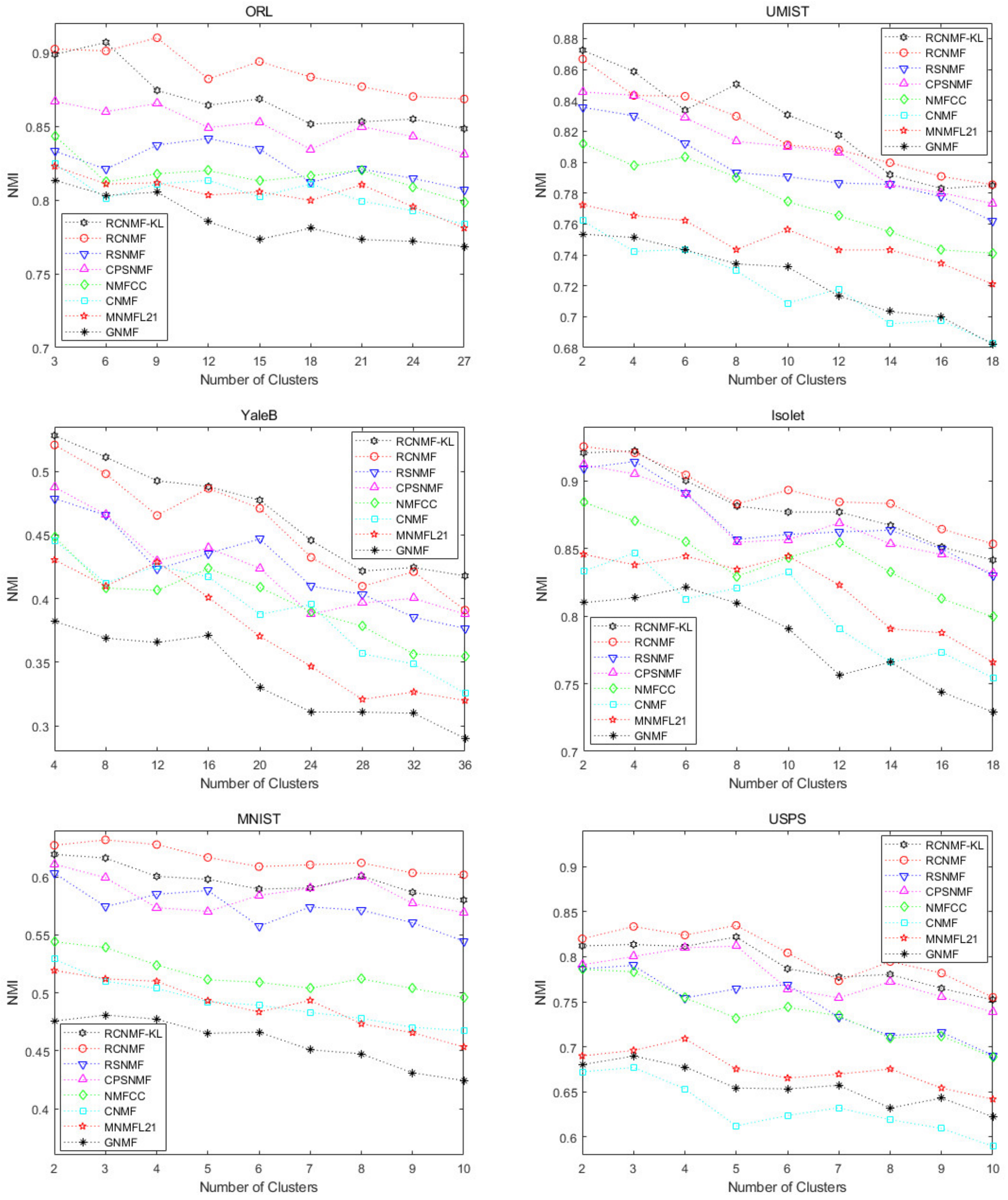

Figure 2. Clustering results by the NMI measurement with different numbers of clusters.

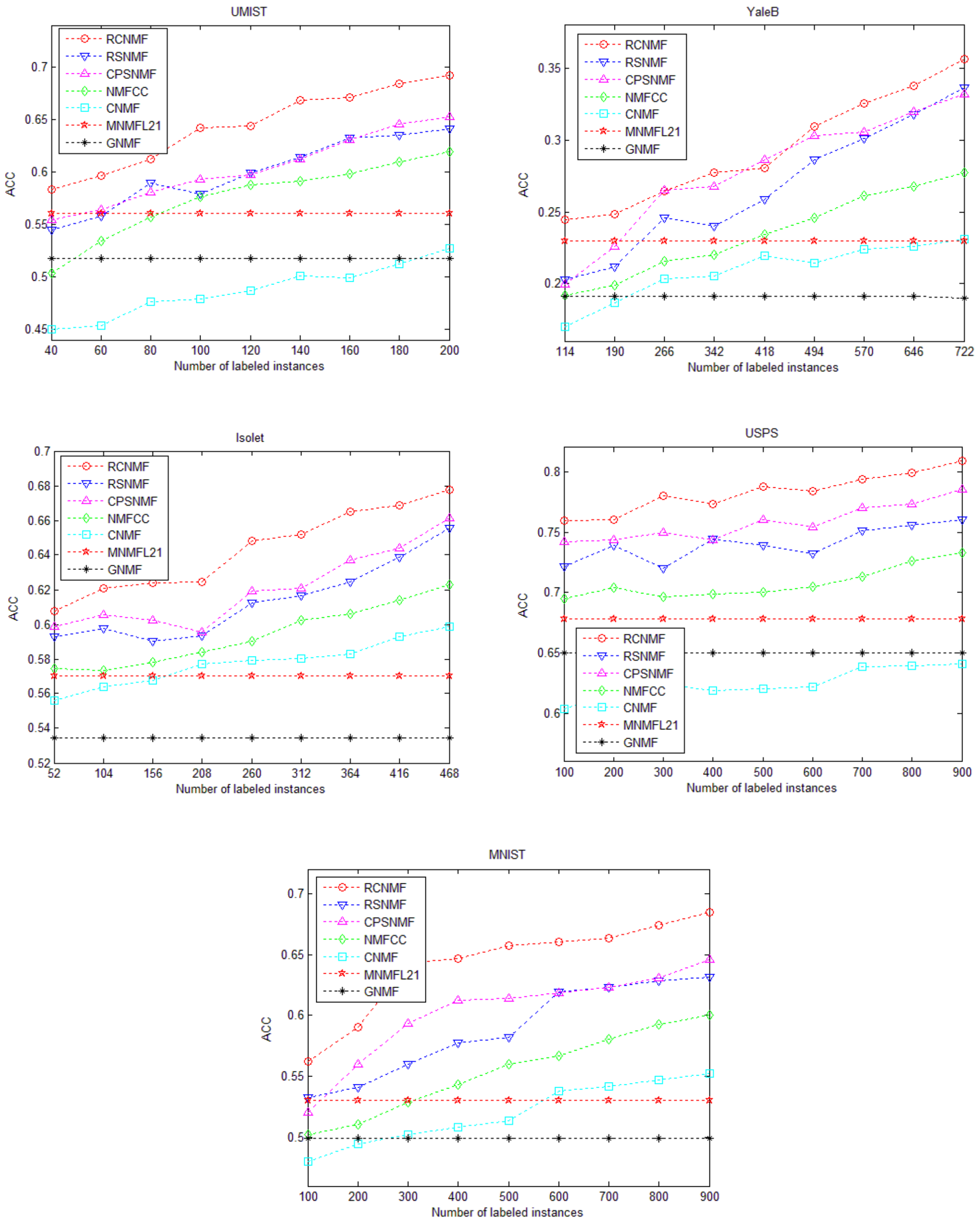

Figure 3. Clustering results by the ACC measurement with different numbers of labeled instances.

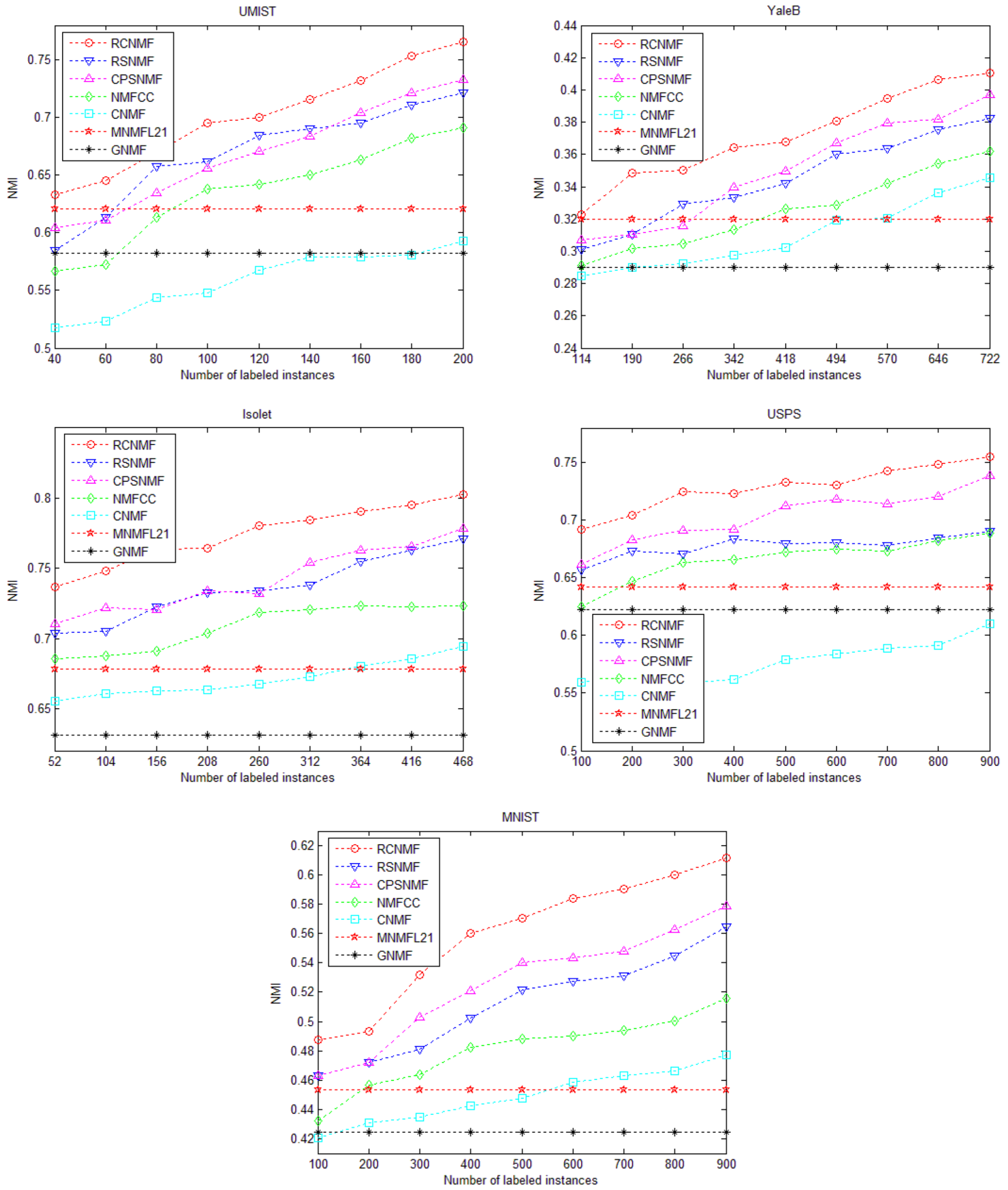

Figure 4. Clustering results by the NMI measurement with different numbers of labeled instances.

1) The proposed RCNMF algorithm can provide better performance on the given six data sets and is superior to other algorithms. This is due to the fact that our RCNMF can learn a compact and discriminating representation. Besides, the RCNMF is the only one to achieve high performance on all the data sets used by this paper. Obviously, local and global structures of data via constraint propagation play a significant role in formulating the intrinsic data representation. Therefore, the RCNMF is able to effectively apply the pairwise constraints to promote the performance of NMF-based algorithm.

2) RCNMF, RSNMF, CPSNMF, NMFCC and CNMF are consistently superior to the baseline method on almost data sets. This indicates that semi-supervised NMF learning from a combination of both labeled and unlabeled instances can guide matrix decomposition better than unsupervised peer. Interestingly, these semi-supervised algorithms have significant differences on the clustering performance. For example, for the MNIST data set, the proposed RCNMF method achieves the highest

clustering performance of 68.45% by ACC, while CNMF just achieves 55.23%. For two unsupervised methods, as a sparse version of GNMF, MNMFL$_{21}$ always achieves better performance than GNMF.
3) The results show that RCNMF outperforms CPSNMF, although both RCNMF and CPSNMF use constraint propagation to discover discriminating representations for data. This is due to two reasons. One is that the L$_{2,1}$-norm objective function of RCNMF is robust to noise and outliers. The loss function using L$_{2,1}$-norm can substantially improve the performance of NMF, which is testified by NMF-based methods [8,56,58,59]. The other is that the L$_{2,1}$-norm regularization term on the basis matrix U in RCNMF can guarantee U sparse in rows and selects some representative features. The sparse formulation can bring encouraging performance improvements. Additionally, it is easy to get the optimal solution for RCNMF because of the additional orthonormal constraint on the factor V in the objective function.
4) CPSNMF outperforms other three semi-supervised methods on most data sets. RSNMF embeds instances from the same class into the same subspace by exploiting the block-diagonal structure of data. NMFCC enforces a cannot-link to approximate 0 and a must-link to approximate 1, which is added to NMF as the regularization term. CNMF forces instances with the same label to have the same representation in the low-dimensional space. However, RSNMF, NMFCC and CNMF ignore the important role of the local structure of the data space for steering matrix decomposition. Besides, the three methods do not map instances from different classes sufficient far in the low-dimensional representation space. We can observe that NMFCC and CNMF perform relatively poorly among the five semi-supervised NMF methods. In addition to the above shortcomings, another disadvantage is that both NMFCC and CNMF are sensitive to data noise and outliers.
5) Another interesting observation is that semi-supervised NMF methods achieve relatively low clustering accuracy when a few labeled instances from data sets are chosen in the experiment. For example, on UMIST, YaleB and MNIST data set, MNMFL$_{21}$ is comparable with CPSNMF, RSNMF, NMFCC and CNMF. Few labeled instances cannot represent the data distribution well when the data set distribution is complex. In fact, two or three labeled instances of each class are not enough to characterize a data representation. However, as the number of available labeled instances, the performance of semi-supervised methods is steadily improved. The propose method can bring encouraging performance improvements compared to other semi-supervised and unsupervised NMF methods, when there are a small number of labeled instances. The reason is that RCNMF efficiently use the supervised information and sufficiently maps instances from different clusters far away from each other in the low-dimensional space.

5.5. Robustness investigation

To investigate the robustness of these methods, we add salt & pepper noise with different densities to four data sets used in this paper. It is worth noting that the data with noise density of 0 is clean data. The experimental results shown in Figures 5 and 6. It can be observed that the performance of all algorithms decreases when the data are contaminated by noise. Moreover, the higher the noise density, the more performance degradation of the algorithm. Although the performance of the proposed RCNMF degrades when decomposing noisy data, it still outperforms the other compared algorithms. As we can see, RCNMF-KL can also obtain relatively satisfactory results. Compared with RCNMF, its performance decreases more with the increase of noise density of the data. MNMFL$_{21}$ and RSNMF are superior to the remaining algorithms. This is due to a fact that the L$_{2,1}$ norm-based cost function is robust to noise and outlier.

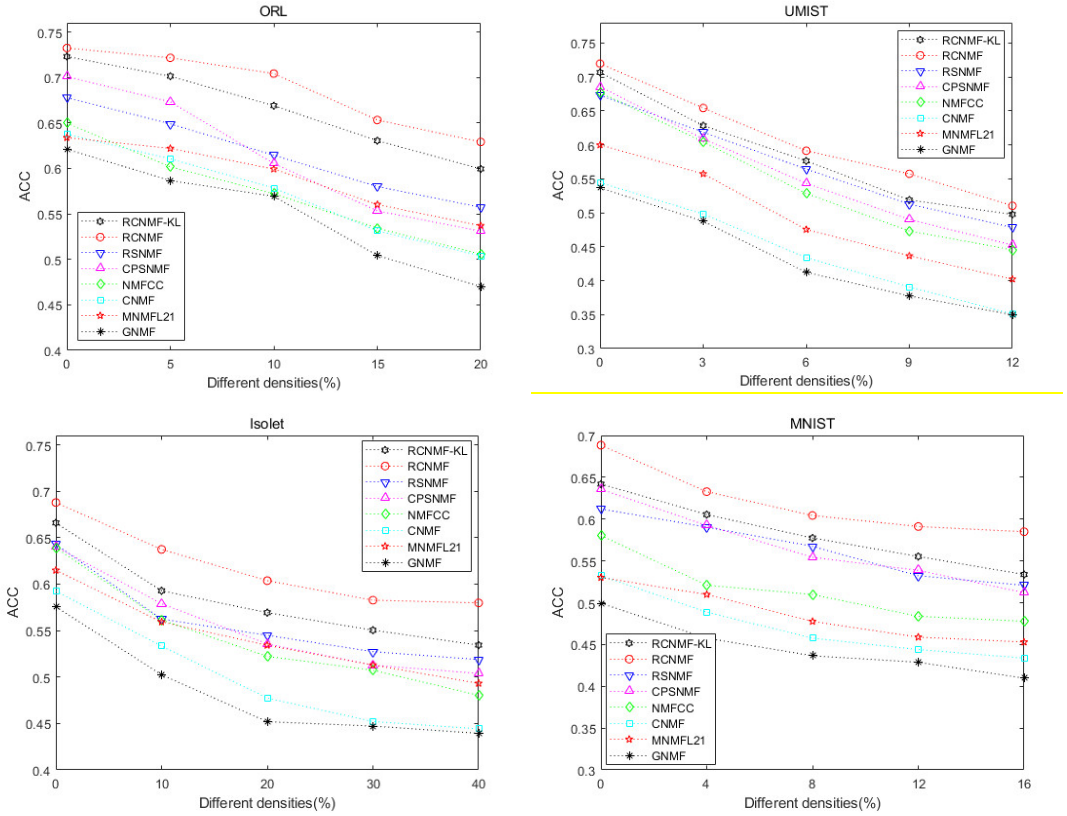

Figure 5. Clustering performance (%) evaluated by ACC with different densities of noise.

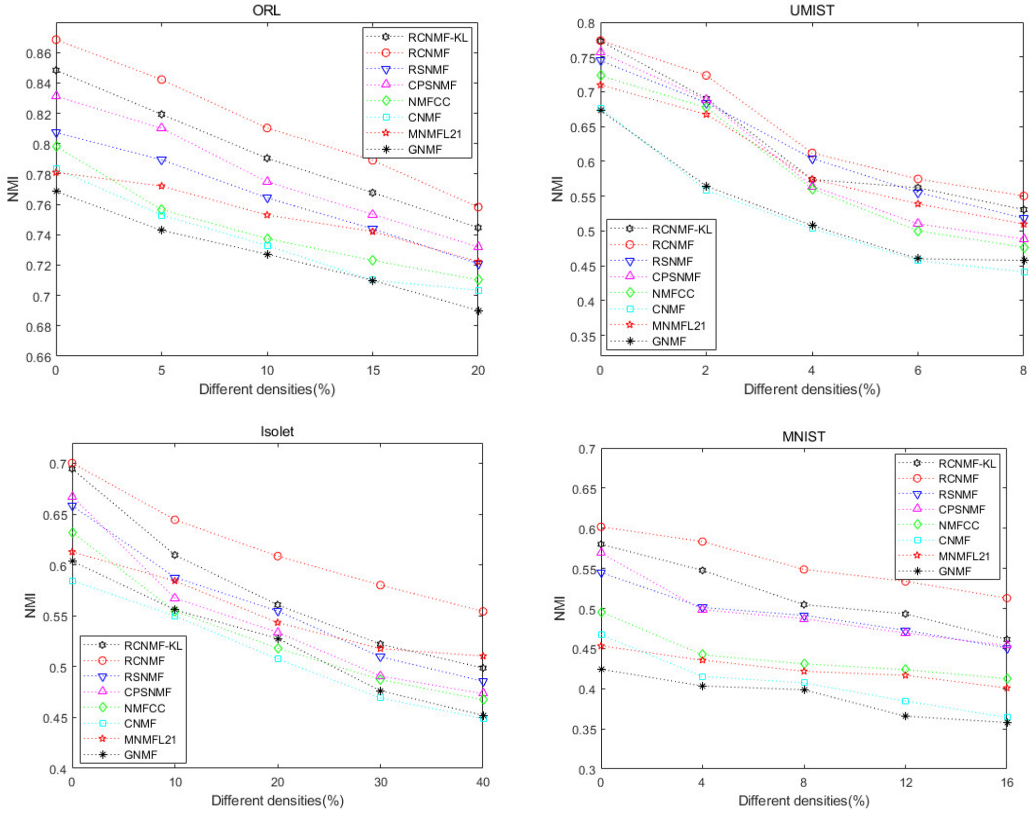

Figure 6. Clustering performance (%) evaluated by NMI with different densities of noise.

5.6. Parameter selection

The RCNMF method has three necessary parameters: nearest neighbors p, two trade-off parameters α and β. The proposed algorithm will degenerate into the $L_{2,1}$-norm NMF ($L_{2,1}$-NMF) [58] when $\alpha = 0$ and $\beta = 0$. Our RCNMF becomes the $L_{2,1}$ norm version of CPSNMF when $\beta = 0$. Figure 7 demonstrates how the average accuracy of the RCNMF becomes with the trade-off parameters α and β, respectively. We can see that different parameters have different effects on the performance of the algorithm. When these two parameters are in the range of [0.1, 100], RCNMF can achieve consistently good performance.

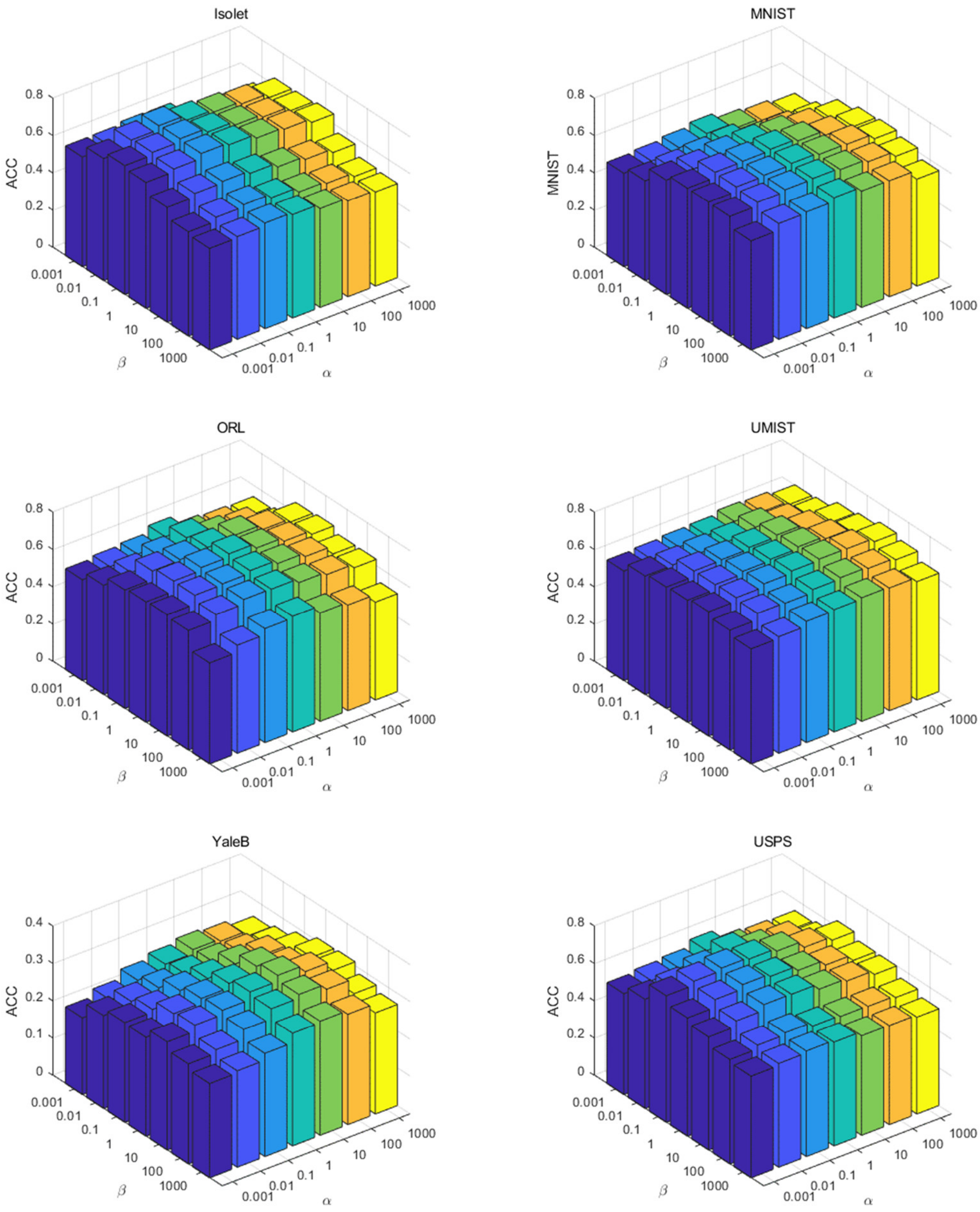

Figure 7. ACC of RCNMF with different α and β on six data sets.

Figure 8 shows that the performance of RCNMF becomes with the nearest neighbor parameter *k*.

As we can see, different values of k have great influence on the RCNMF. Graph-based methods generally construct k-nearest graph to depict the local structure of the data space. These methods are based on the assumption that nearby instances have the same class label [9,15,21–23,35,55,56]. Obviously, this assumption likely fails to hold when k enlarges. This is why the performance of RCNMF drops with the increase of k. In fact, graph-based methods suffer from this torment as reported in [23,35]. Generally, the value of p ranges from an integer of 3 to 9.

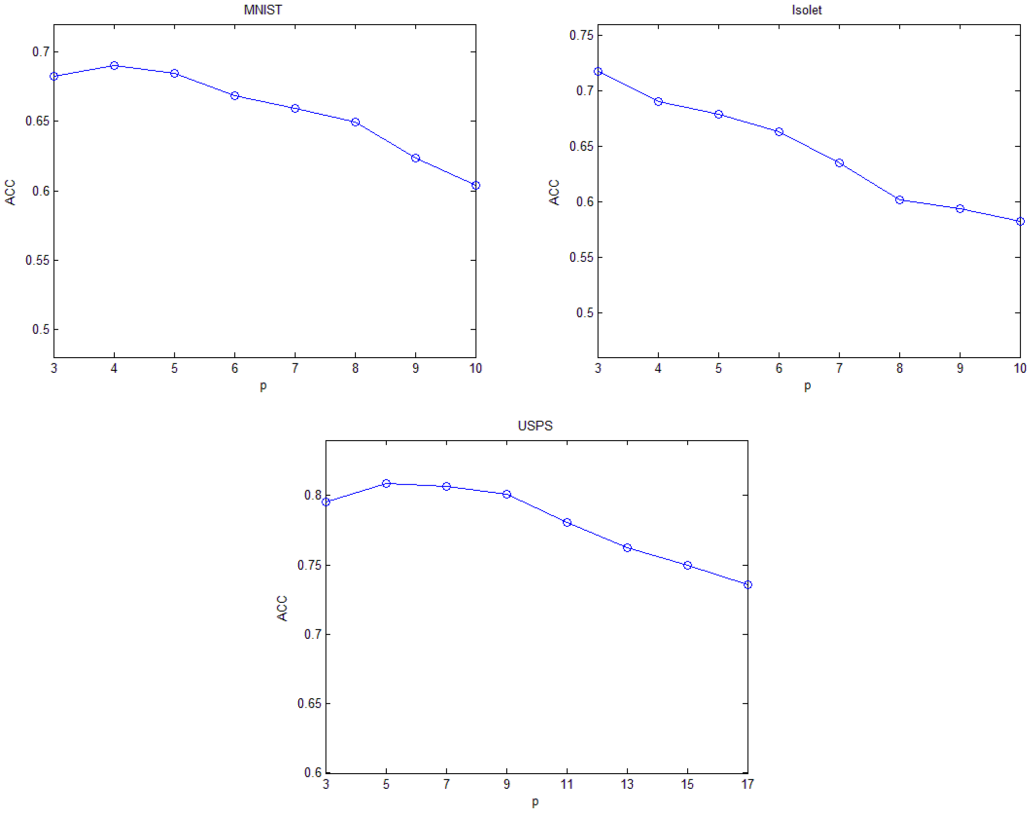

Figure 8. The accuracy of RCNMF versus parameter p.

5.7. Convergence study

In Section 4, the convergence of the proposed method has been proven and also the computational complex is analyzed. Here, we inspect the convergence speed of the proposed algorithm. Figure 9 demonstrates the convergence cures of RCNMF on the two data sets. In every figure, the x-axis represents the iteration number and the y-axis describes the value of the cost function. We can observe that the updating rule of the proposed RCNMF method converge relatively fast. Generally, the iterations of our algorithm on these data sets are less than 300.

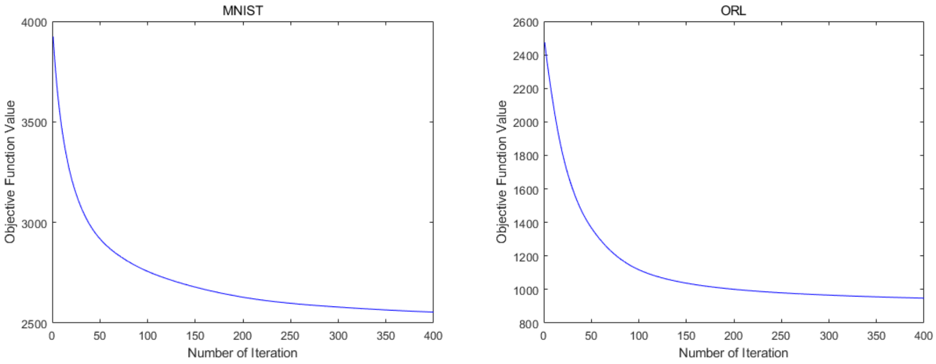

Figure 9. Convergence curves of the proposed RCNMF method.

6. Conclusions

We have proposed a novel robust semi-supervised nonnegative matrix factorization algorithm, called RCNMF. RCNMF models local and global structures of data via constraint propagation to make latent representations of instances involved by the same class are mapped closer and those of instances involved by different classes farther. The proposed method introduces the $L_{2,1}$-norm into the objective function and thus is robust to noise and outliers. Furthermore, the $L_{2,1}$-norm constraint for the factorial matrix is added to the loss function as the regularizer, which ensures the new representation sparse in rows. Experimental results on six real-world data sets show that our proposed framework is superior to other state-of-the-art algorithms.

Acknowledgments

This work was supported by Industry & University Research Cooperation Project of Jiangsu Province under Grant BY2020056. The authors extend their appreciation to the Deanship of Scientific Research at King Saud University for funding this work through research group no. RG-1441-331.

Conflict of interest

The authors declare no conflict of internet.

References

1. S. Liu, L. Liu, J. Tang, B. Yu, Y. Wang, W. Shi, Edge computing for autonomous driving: Opportunities and challenges, in *Proceedings of the IEEE*, **107** (2019), 1697–1716. doi: 10.1109/JPROC.2019.2915983.
2. M. Wang, X. Hua, J. Tang, R. Hong, Beyond distance measurement: constructing neighborhood similarity for video annotation, *IEEE Trans. Multimedia*, **11** (2009), 465–476. doi: 10.1109/TMM.2009.2012919.
3. Y. Song, W. Cai, H. Huang, D. Feng, Y. Wang, M. Chen, Bioimage classification with subcategory discriminant transform of high dimensional visual descriptors, *BMC Bioinf.*, **17** (2016), 465. doi: 10.1186/s12859-016-1318-9.

4. Z. Xing, Y. Ma, X. Yang, F. Nie, Graph regularized nonnegative matrix factorization with label discrimination for data clustering, *Neurocomputing*, **440** (2021), 297–309. doi: 10.1016/j.neucom.2021.01.064.
5. H. Xiong, D. Kong, Elastic nonnegative matrix factorization, *Pattern Recognit.*, **90** (2019), 464–475. doi: 10.1016/j.patcog.2018.07.007.
6. F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via Joint ℓ2, 1-norms minimization, in *Proceedings of the 23rd International Conference on Neural Information Processing Systems (NIPS)*, **2** (2010), 1813–1821. doi: 10.5555/2997046.2997098.
7. R. Chatpatanasiri, B. Kijsirikul, A unified semi-supervised dimensionality reduction framework for manifold learning, *Neurocomputing*, **73** (2010), 1631–1640. doi: 10.1016/j.neucom.2009.10.024.
8. Z. Li, J. Tang, X. He, Robust structured nonnegative matrix factorization for image representation, *IEEE Trans. Neural Networks Learn. Syst.*, **29** (2018), 1947–1960. doi: 10.1109/TNNLS.2017.2691725.
9. W. Yu, R. Wang, F. Nie, F. Wang, Q. Yu, X. Yang, An improved locality preserving projection with l1-norm minimization for dimensionality reduction, *Neurocomputing*, **316** (2018), 322–331. doi: 10.1016/j.neucom.2018.08.008.
10. P. N. Belhumeur, J. P. Hepanha, D. J. Kriegman, Eigenfaces vs. fisherfaces: recognition using class specific linear projection, *IEEE Trans. Pattern Anal. Mach. Intell.*, **19** (1997), 711–720. doi: 10.1109/34.598228.
11. S. Yan, D. Xu, B. Zhang, H. Zhang, Q. Yang, S. Lin, Graph embedding and extensions: A general framework for dimensionality reduction, *IEEE Trans. Pattern Anal. Mach. Intell.*, **29** (2007), 40–51. doi: 10.1109/TPAMI.2007.250598.
12. S. Roweis, L. K. Saul, Nonlinear dimensionality reduction by locally linear embedding, *Science*, **290** (2000), 2323–2326. doi: 10.1126/science.290.5500.2323.
13. J. B. Tenenbaum, V. Silva, J. C. Langford, A global geometric framework for nonlinear dimensionality reduction, *Science*, **290** (2000), 2319–2323. doi: 10.1126/science.290.5500.2319.
14. A. M. Martinez, A. C. Kak, PCA versus LDA, *IEEE Trans. Pattern Anal. Mach. Intell.*, **23** (2001), 228–233. doi: 10.1109/34.908974.
15. F. Nie, D. Xu, I. W. Tsang, C. Zhang, Flexible manifold embedding: A framework for semi-supervised and unsupervised dimension reduction, *IEEE Trans. Image Process.*, **19** (2010), 1921–1932. doi: 10.1109/TIP.2010.2044958.
16. D. Zhang, Z. Zhou, S. Chen, Semi-supervised dimensionality reduction, in *Peoceedings of the 2007 SIAM International Conference on Data Mining (SDM)*, (2007), 629–634. doi: 10.1137/1.9781611972771.73.
17. C. Boutsidis, P. Drineas, M. W Mahoney, P. Drineas, Unsupervised feature selection for the k-means clustering problem, in *Proceedingds of the 22nd International Conference on Neural Information Processing Systems*, (2009), 153–161. doi: 10.5555/2984093.2984111.
18. D. Cai, X. He, J. Han, Semi-supervised discriminant analysis, in *2007 IEEE 11th International Conference on Computer Vision (ICCV)*, (2007), 1–7. doi: 10.1109/ICCV.2007.4408856.
19. J. Ye, R. Janardan, C. Park, H. Park, An optimization criterion for generalized discriminant analysis on under sampled problems, *IEEE Trans. Pattern Anal. Mach. Intell.*, **26** (2004), 982–994. doi: 10.1109/TPAMI.2004.37.

20. X. Wang, Y. Liu, F. Nie, H. Huang, Discriminative unsupervised dimensionality reduction, in *Proceedings of the 24th International Conference on Artificial Intelligence*, (2015), 3925–3931. doi: 10.5555/2832747.2832796.
21. X. He, P. Niyogi, Locality preserving projections, in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, (2003), 153–160. doi: 10.5555/2981345.2981365.
22. M. Belkin, P. Niyogi, Laplacian eigenmaps and spectral techniques for embedding and clustering, in *Proceedings of the 14th International Conference on Neural Information Processing System*, (2001), 585–591. doi: 10.5555/2980539.2980616.
23. D. Wang, X. Gao, X. Wang, Semi-supervised nonnegative matrix factorization via constraint propagation, *IEEE Trans. Cybern.*, **46** (2016), 233–244. doi: 10.1109/TCYB.2015.2399533.
24. D. D. Lee, H. S. Seung, Learning the parts of objects by non-negative matrix factorization, *Nature*, **401** (1999), 788–791. doi: 10.1038/44565.
25. Y. X. Wang, Y. J. Zhang, Nonnegative matrix factorization: a comprehensive review, *IEEE Trans. Knowl. Data Eng.*, **25** (2013), 1336–1353. doi: 10.1109/TKDE.2012.51.
26. S. Li, X. Hou, H. Zhang, Q. Cheng, Learning spatially localized, parts-based representation, in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2001), 1. doi: 10.1109/CVPR.2001.990477.
27. S. D. Kamvar, D. Klein, C. D. Manning, Spectral learning, in *Proceedings of the 18th International Joint Conference on Artificial Intelligence*, (2003), 561–566. doi: 10.5555/1630659.1630742.
28. Q. Huang, X. Yin, S. Chen, Y. Wang, B. Chen, Robust nonnegative matrix factorization with structure regularization, *Neurocomputing*, **412** (2020), 72–90. doi: 10.1016/j.neucom.2020.06.049.
29. S. Y. Li, Y. Jiang, Z. H. Zhou, Partial multi-view clustering, in *Proceedings of the 28th AAAI Conference on Artificial Intelligence*, (2014), 1968–1974. doi: 10.5555/2892753.2892826.
30. Y. Yi, J. Wang, W. Zhou, C. Zheng, J. Kong, S. Qiao, Non-negative matrix factorization with locality constrained adaptive graph, *IEEE Trans. Circuits Syst. Video Techn.*, **30** (2020), 427–441. doi: 10.1109/TCSVT.2019.2892971.
31. H. Liu, Z. Wu, X. Li, D. Cai, T. S. Huang, Constrained nonnegative matrix factorization for image representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **34** (2012), 1299–1311. doi: 10.1109/TPAMI.2011.217.
32. J. P. Brunet, P. Tamayo, T. R. Golub, J. P. Mesirov, Metagenes and molecular pattern discovery using matrix factorization, in *Proceedings of the National Academy of Sciences*, **101** (2004), 4164–4169. doi: 10.1073/pnas.0308531101.
33. C. Peng, Y. Chen, Z. Kang, C. Chen, Q. Cheng, Robust principal component analysis: A factorization-based approach with linear complexity, *Inf. Sci.*, **513** (2020), 581–599. doi: 10.1016/j.ins.2019.09.074.
34. C. Ding, T. Li, M. I. Jordan, Convex and semi-nonnegative matrix factorizations, *IEEE Trans. Pattern Anal. Mach. Intell.*, **32** (2010), 45–55. doi: 10.1109/TPAMI.2008.277.
35. D. Cai, X. He, J. Han, T. S. Huang, Graph regularized nonnegative matrix factorization for data representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **33** (2011), 1548–1560. doi: 10.1109/TPAMI.2010.231.
36. Z. Li, J. Liu, H. Lu, Structure preserving non-negative matrix factorization for dimensionality reduction, *Comput. Vision Image Understanding*, **117** (2013), 1175–1189. doi: 10.1016/j.cviu.2013.04.003.

37. Z. Zhang, K. Zhao, Low rank matrix approximation with manifold regularization, *IEEE Trans. Pattern Anal. Mach. Intell.*, **35** (2013), 1717–1729. doi: 10.1109/TPAMI.2012.274.
38. N. Lu, H. Miao, Structure constrained nonnegative matrix factorization for pattern clustering and classification, *Neurocomputing*, **171** (2016), 400–411. doi: 10.1016/j.neucom.2015.06.049.
39. A. Cichocki, R. Zdunek, S. Amari, Csiszár's divergences for non-negative matrix factorization: Family of new algorithms, *ICA Independent Component Analysis and Blind Signal Separation*, (2006), 32–39. doi: 10.1007/11679363_5.
40. A. Cichocki, R. Zdunek, A. Phan, S. Amari, Nonnegative matrix and tensor factorizations: applications to exploratory multi-way data analysis and blind source separation, *John Wiley & Sons, Ltd.*, (2009). doi:10.1002/9780470747278.
41. C. Fevotte, J. Idier, Algorithms for nonnegative matrix factorization with the β-divergence, *Neural Comput.*, **23** (2011), 2421–2456. doi: 10.1162/NECO_a_00168.
42. K. Devarajan, V. C. K. Cheung, A quasi-likelihood approach to nonnegative matrix factorization, *Neural Comput.*, **28** (2016), 1663–1693. doi: 10.1162/NECO_a_00853.
43. K. Devarajan, A statistical framework for non-negative matrix factorization based on generalized dual divergence, *Neural Networks*, **140** (2021), 309–324. doi: 10.1016/j.neunet.2021.03.020.
44. Y. Chen, M. Rege, M. Dong, J. Hua, Non-negative matrix factorization for semi-supervised data clustering, *Knowl. Inf. Syst.*, **17** (2008), 355–379. doi: 10.1007/s10115-008-0134-6.
45. N. Guan, X. Huang, L. Lan, Z. Luo, X. Zhang, Graph based semi-supervised non-negative matrix factorization for document clustering, in *2012 11th International Conference on Machine Learning and Applications (ICMLA)*, (2012), 404–408. doi: 10.1109/ICMLA.2012.73.
46. H. Lee, J. Yoo, S. Choi, Semi-supervised nonnegative matrix factorization, *IEEE Signal Process. Lett.*, **17** (2010), 4–7. doi: 10.1109/LSP.2009.2027163.
47. X. Zhang, L. Zong, X. Liu, J. Luo, Constrained clustering with nonnegative matrix factorization, *IEEE Trans. Neural Networks Learn. Syst.*, **27** (2016), 1514–1526. doi: 10.1109/TNNLS.2015.2448653.
48. Z. Yang, Y. Xiang, K. Xie, Y. Lai, Adaptive method for nonsmooth nonnegative matrix factorization, *IEEE Trans. Neural Networks Learn. Syst.*, **28** (2017), 948–960. doi: 10.1109/TNNLS.2016.2517096.
49. Y. Yi, Y. Shi, H. Zhang, J. Wang, Jun Kong, Label propagation based semi-supervised non-negative matrix factorization for feature extraction, *Neurocomputing*, **149** (2015), 1021–1037. doi: 10.1016/j.neucom.2014.07.031.
50. Y. Yi, Y. Chen, J. Wang, G. Lei, J. Dai, H. Zhang, Joint feature representation and classification via adaptive graph semi-supervised nonnegative matrix factorization, *Signal Process.: Image Commun.*, **89** (2020), 115984. doi: 10.1016/j.image.2020.115984.
51. S. Li, Q. Liu, J. Dai, W. Wang, X. Gui, Y. Yi, Adaptive-weighted multiview deep basis matrix factorization for multimedia data analysis, *Wireless Commun. Mobile Comput.*, **9** (2021), 1–12. doi: 10.1155/2021/5526479.
52. Y. Jia, S. Kwong, J. Hou, W. Wu, Semi-supervised non-negative matrix factorization with dissimilarity and similarity regularization, *IEEE Trans. Neural Networks Learn. Syst.*, **31** (2019), 2510–2521. doi: 10.1109/TNNLS.2019.2933223.
53. Z. Xing, M. Wen, J. Peng, J. Feng, Discriminative semi-supervised non-negative matrix factorization for data clustering, *Eng. Appl. Artif. Intell.*, **103** (2021), 104289. doi: 10.1016/j.engappai.2021.104289.

54. D. Zhou, O. Bousquet, T. N. Lal, J. Weston, B. S. Cholkopf, Learning with local and global consistency, in *Proceedings of the 16th International Conference on Neural Information Processing Systems*, (2003), 321–328. doi: 10.5555/2981345.2981386.
55. Z. Li, J. Liu, J. Tang, H. Lu, Robust structured subspace learning for data representation, *IEEE Trans. Pattern Anal. Mach. Intell.*, **37** (2015), 2085–2098. doi: 10.1109/TPAMI.2015.2400461.
56. J. Huang, F. Nie, H. Huang, C. Ding, Robust manifold nonnegative matrix factorization, *ACM Trans. Knowl. Discovery Data*, **8** (2014), 1–21. doi: 10.1145/2601434.
57. W. Liu, N. Zheng, Q. You, Nonnegative matrix factorization and its applications in pattern recognition, *Chin. Sci. Bull.*, **51** (2006), 7–18. doi: 10.1007/s11434-005-1109-6.
58. D. Kong, C. Ding, H. Huang, Robust nonnegative matrix factorization using l21 norm, in *Proceedings of the 20th ACM International Conference on Information and Knowledge Management*, (2011), 673–682. doi: 10.1145/2063576.2063676.
59. B. Wu, E. Wang, Z. Zhu, W. Chen, P. Xiao, Manifold NMF with $L_{2,1}$ norm for clustering, *Neurocomputing*, **273** (2018), 78–88. doi: 10.1016/j.neucom.2017.08.025.
60. M. Babaee, S. Tsoukalas, M. Babaee, G. Rigoll, M. Datcu, Discriminative nonnegative matrix factorization for dimensionality reduction, *Neurocomputing*, **173** (2016), 212–223. doi: 10.1016/j.neucom.2014.12.124.
61. Z. Lu, Y. Peng, Exhaustive and efficient constraint propagation: A graph-based learning approach and its applications, *Int. J. Comput. Vision*, **103** (2013), 306–325. doi: 10.1007/s11263-012-0602-z.
62. C. Ding, D. Zhou, X. He, H. Zha, R1-pca: Rotational invariant l1-norm principal component analysis for robust subspace factorization, in *Proceedings of the 23rd International Conference on Machine Learning (ICML)*, (2006), 281–288. doi: 10.1145/1143844.1143880.
63. X. Yin, S. Chen, E. Hu. Regularized soft K-means for discriminant analysis, *Neurocomputing*, **103** (2013), 29–42. doi: 10.1016/j.neucom.2012.08.021.