Neural Processing Letters (2022) 54:5721–5739
https://doi.org/10.1007/s11063-022-10882-x

Adaptive Graph Regularized Deep Semi-nonnegative Matrix Factorization for Data Representation

Zhenqiu Shu[1] · Yanwu Sun[2] · Jiali Tang[2] · Congzhe You[2]

Accepted: 7 May 2022 / Published online: 28 May 2022

Abstract
Recently, matrix factorization-based data representation methods exhibit excellent performance in many real applications. However, traditional deep semi-nonnegative matrix factorization (DSNMF) models the relationship between samples by predefining a fixed graph, which is not optimal and thus cannot exploit the intrinsic local structure among data effectively. In this work, an adaptive graph regularized deep semi-nonnegative matrix factorization (AGRDSNMF) algorithm is proposed for data representation. This proposed AGRDSNMF method can construct an adaptive optimal graph in each layer, whose weights are automatically determined by the probabilities between neighborhood samples. Then the adaptive graph regularizer of each layer is adopted to constrain the corresponding coefficient matrix during decomposition. Therefore, AGRDSNMF can capture the geometric structure of the representation in each layer. Experiments are conducted on COIL20, PIE, and TDT2 datasets, and our AGRDNSMF algorithm can achieve encouraging clustering performance.

Keywords Matrix factorization · Data representation · Graph regularized · Deep semi-NMF · Adaptive graph · Geometric structure · Optimal graph · Clustering

1 Introduction

Data representation is a typical problem in machine learning and pattern recognition. In many fields, the real data tend to exist with high-dimensionality. Therefore, traditional clustering and classification methods need computationally expensive to deal with high-dimensional data. Matrix factorization techniques show its effectiveness and efficiency in high-dimensional data processing. In the past few decades, it is an active research topic that the matrix factorization techniques are applied to explore the high-level semantic information hidden in data. Nowadays, matrix factorization techniques are widely applied to face recognition [1, 2], cross-modal retrieval [3–5], hyperspectral unmixing [6, 7] document clustering

✉ Zhenqiu Shu
shuzhenqiu@126.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

[8, 9], tumor classification [10, 11], etc. In recent years, researchers have still proposed many variants of matrix factorization methods for data representation [12, 13].

Nonnegative matrix factorization (NMF) [14] is a widely-used data representation method due to its strong support in perception science. It aims to linearly represent the original nonnegative data matrix with a set of basis vectors. Moreover, their updating rules perform only additive, not subtractive, operations due to nonnegativity constraint. Therefore, it can learn the part attributes of data. However, the nonnegativity constraint is too strict to handle the real data mixing with negative elements. To address this issue, Ding et al. [15] developed a novel Semi-NMF (SNMF) method for data clustering. It relaxes the nonnegative constraints of factorizations and only limits the elements in the coefficient matrix to be nonnegative. Therefore, SNMF can handle the data with negative components. Xu et al. [16] put forward a concept factorization (CF) approach that not only deals with a data matrix containing the negative elements, but also easily handles nonlinear data by adopting kernel trick. In recent years, various constraints have been applied to further improve the representation power of algorithms in real problems. To discover the manifold structure embedded in the data, Cai et al. [17] put forward a graph regularized NMF (GNMF) algorithm that tries to model the geometric manifold structure of data by constructing a nearest neighbor graph. Shang et al. [18] further introduced a dual graph regularized NMF method that takes full advantage of the manifold structure in both feature and data space using two graph regularizers. Shu et al. [19] proposed adding the rank constraint to the Laplacian matrix of the graph, and then incorporated it into the traditional NMF framework. In RCNMF, the rank constraint is imposed on the Laplacian matrix of the graph model. Thus, it makes sure to get k connected components that are more conducive to clustering.

Inspired by multilayer architecture, Rajabi et al. [20] recently introduced a multilayer NMF (MLNMF) method, in which the observation matrix is iteratively decomposed into several layers. Moreover, the sparseness constraint is imposed on both the signature matrix and the abundance matrix. Shu et al. [21] further considered the dual manifold structure of hyperspectral image using a dual graph regularizer based multilayer structure. Tong et al. [22] integrated an adaptive graph regularizer into the model of MLNMF instead of a predefined fixed graph regularizer. However, its disadvantage is that an adaptive graph constructed in high-dimensional space is used to preserve the manifold structure of all presentation layers. Therefore, it cannot effectively discover the manifold structure of each layer. Motived by the success of deep learning, Fang et al. [23] developed a sparsity-constrained deep NMF algorithm by imposing the l_1 norm sparse constraint on the abundance matrix in each layer. To consider the hierarchical features with hidden information, Feng et al. [24] further promoted the piecewise smoothness of the abundance matrix using the TV regularizer. Zhao et al. [25] proposed a novel deep non-negative basis matrix factorization (DNBMF) method. Different from the above-mentioned methods, DNBMF tries to learn the underlying basics by decomposing the basis matrix layer by layer. Trigeorgis et al. [26] put forward a deep semi-nonnegative matrix factorization (DSNMF) algorithm, in which a hierarchy of attributes hidden in data are effectively learned by constructing a deep framework. Then the semi-supervised learning extension of DSNMF was further proposed to utilize the known prior information among data using a fixed graph regularizer.

In this work, we propose an adaptive graph regularized deep semi-NMF (AGRDSNMF) algorithm to represent the high-dimensional data. AGRDSNMF adopts the deep architecture to discover the hidden features of data. Different from the fixed-graph based NMF method, the proposed AGRDSNMF method tries to learn an optimal adaptive graph in each layer based on probability theory. Therefore, in AGRDSNMF, the relationship of the coefficient distribution in each layer is effectively exploited. Besides, an effective and efficient strategy is applied

to optimize the objective function. Experimental results on different benchmark datasets manifest that our AGRDSNMF algorithm significantly outperforms other competitors in clustering.

The rest of this paper is organized as follows. Section II introduces the related works of the proposed method. Section III describes our proposed AGRDSNMF method. The experimental results are shown in Section IV, and the conclusions are drawn in Section V.

2 Related Works

2.1 NMF

Assuming that a data matrix $X \in R^{m \times n}$, and for each element x_{ij} in X,$x_{ij} \geq 0$ is always required. NMF aims to approximate the observation matrix using the product of two low-rank nonnegative matrices U and V. Therefore, NMF seeks a decomposition model as follows:

$$X = UV, \tag{1}$$

where U and V are regarded as the basis matrix and the coefficient matrix, respectively. Therefore, the model of NMF is expressed by the following minimization problem:

$$\begin{aligned} O &= ||X - UV||_F^2 \\ s.t.\, &U \geq 0, V \geq 0, \end{aligned} \tag{2}$$

where $||\cdot||_F$ stands for the Frobenius norm. An efficient multiplicative iterative scheme [14] is developed to optimize the problem (2). Therefore, the updating rules of Eq. (2) are derived as follows:

$$U_{ij} \leftarrow U_{ij} \frac{\left(XV^T\right)_{ij}}{\left(UVV^T\right)_{ij}}, \; V_{ij} \leftarrow V_{ij} \frac{\left(U^T X\right)_{ij}}{\left(U^T UV\right)_{ij}}. \tag{3}$$

2.2 SNMF

In many real applications, the original data matrix contains some negative elements due to the noise. Traditional NMF methods cannot deal with this data matrix due to its strict nonnegative constraint. To overcome this drawback, Chris et al. [15] proposed a SNMF method that relaxes the nonnegative constraint on both the data matrix and the basis matrix, and only constrains the coefficient matrix nonnegative. Therefore, it aims to seek a basis matrix and a nonnegative coefficient matrix, whose product approximates the data matrix as follows:

$$X^{\pm} \approx U^{\pm} V^{+}. \tag{4}$$

In fact, some studies have shown that SNMF is equal to k-means while each column contains only one positive element [15]. The model of SNMF can be rewritten the following minimization problem:

$$\begin{aligned} O_{SNMF} &= ||X^{\pm} - U^{\pm} V^{+}||_F^2 \\ s.t.\, &V \geq 0, \end{aligned} \tag{5}$$

where $||\cdot||_F$ is the Frobenius norm. Using a similar strategy, the solution of the formula (5) can be obtained using the following rules:

$$U \leftarrow XV^{\dagger}, \tag{6}$$

$$V \leftarrow V\sqrt{\frac{\left(X^TU\right)^+ + V^T\left(U^TU\right)^-}{\left(X^TU\right)^- + V^T\left(U^TU\right)^+}}, \tag{7}$$

where † denotes Moore–Penrose pseudoinverse. $(\cdot)^+$ denotes a matrix containing the negative elements replaced by 0, and $(\cdot)^-$ denotes one containing the positive elements replaced by 0.

2.3 DSNMF

Many studies have shown that the deep framework can achieve superior performance in many real applications [27, 28]. Recently, the DSNMF method extends the traditional single-layer decomposition to the deep decomposition [26]. DSNMF can be given as the following form:

$$X^{\pm} = U_1^{\pm}U_2^{\pm}\cdots U_t^{\pm}V_t^{+}.$$

The model of DSNMF is formulated as follows:

$$\begin{aligned} O_{DSNMF} &= ||X - U_1U_2\cdots U_tV_t||_F^2 \\ &= X^TX - 2X^TU_1U_2\cdots U_tV_t + V_t^TU_t^TU_{t-1}^T\cdots U_1^TU_1U_2\cdots U_tV_t. \end{aligned} \tag{8}$$

Therefore, the updating rules of the model (18) is given as follows:

$$\begin{aligned} U_l &= \left(\Phi^T\Phi\right)^{-1}\Phi^TX\tilde{V}_l^T\left(\tilde{V}_l\tilde{V}_l^T\right)^{-1} \\ U_l &= \Phi^{\dagger}X\tilde{V}_l^{\dagger}, \end{aligned} \tag{9}$$

$$V_l = V_l \odot \sqrt{\frac{\left[\Phi^TX\right]^+ + \left[\Phi^T\Phi\right]^-V_l}{\left[\Phi^TX\right]^- + \left[\Phi^T\Phi\right]^+V_l}}, \tag{10}$$

where $\Phi{=}\Pi_{k=1}^{l-1}U_k$.

3 AGRDSNMF

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

3.1 Motivation

In the past few years, the matrix factorization methods based on the deep framework have shown encouraging performance in many applications. However, these methods cannot take full advantage of the prior knowledge hidden in data. Recently, various constrained NMF methods were developed by incorporating prior knowledge to enhance performance [29–31]. To preserve the local geometric structure among data, GNMF was proposed by constructing a predefined fixed graph model [21]. It is obvious that this fixed graph is not optimal and thus cannot effectively explore the intrinsic structure of data. To alleviate these problems mentioned above, we construct a learned adaptive graph to exploit the local neighborhood

relationship of data in each layer. Moreover, the elements of the affinity matrix associated with the adaptive graph are determined by the probability between neighbors. Then these adaptive graphs are integrated into each layer of the deep semi-NMF framework, respectively. Therefore, our proposed AGRDSNMF method not only adopts the deep framework to discover the hidden features, but also fully utilizes the prior knowledge of data.

3.2 Constructing the Adaptive Graph

A given data matrix $X = [x_1, x_2, ..., x_n] \in R^{m\times n}$ includes n samples, where x_i denotes an m-dimensional vector. The weights of two samples x_i and x_j are determined according to their probability. Generally speaking, the smaller the distance between two close data points, the greater the probability assigned to this pair, and vice versa. It is easy to know that the sum of the probability of each sample is one. Therefore, we can give its mathematical form as follows:

$$\min_{p_i^T 1=1, 0\le p_{i\le 1}} \sum_{j=1}^{m} ||x_i - x_j||_2^2 p_{ij}, \tag{11}$$

where p_{ij} denotes the probability of data points x_i and x_j.

Unfortunately, there is a trivial solution to the problem (11), and thus we cannot achieve its optimal solution. In addition, we can see that the problem (11) can be solved by the nearest neighbor of d_i with probability 1 while the probabilities of all the other samples are 0. To solve this issue, a regularization term αp_{ij}^2 is added to the problem (11). Therefore, the problem (11) can be further formulated as follows:

$$\min_{p_i^T 1=1,\, 0\le p_{i\le 1}} \sum_{j=1}^{m} \left(||x_i - x_j||_2^2 p_{ij}^2 + \alpha p_{ij}^2 \right), \tag{12}$$

where α is the regularization parameter. For simplicity, we define $z_i^d \in R^{m\times 1}$ as an m-dimensional vector. Its *j-th* element is denoted as

$$z_{ij}^d = ||x_i - x_j||_2^2. \tag{13}$$

We can use formula (12) to assign the neighbors of each data point x_i. Thus, the neighbors for all samples are determined by solving the following optimization problem:

$$\min_{\forall i, p_i^T 1=1,\, 0\le p_i \le 1} \sum_{i,j=1}^{m} \left(z_{ij}^d p_{ij} + \alpha p_{ij}^2 \right). \tag{14}$$

According to [32], we add the rank constraint to the Laplacian matrix of the graph, and thus we can obtain some connected components of the graph model. The clustering performances can be effectively enhanced by adopting this structure. By imposing the rank constraint, the problem (14) is rewritten as follows:

$$\begin{aligned} &\min_{P} \sum_{i,j=1}^{m} \left(z_{ij}^d p_{ij} + \alpha p_{ij}^2 \right) \\ &s.t. \forall p_i^T 1 = 1,\ 0 \le p_i \le 1,\ rank(L_G) = m - e, \end{aligned} \tag{15}$$

where $P = [p_1, p_2, ..., p_m] \in R^{m\times m}$ is a similarity matrix of the graph on all samples. $D_G \in R^{m\times m}$ is the diagonal matrix of adaptive graph G, whose i-th diagonal element equals $\sum_j (p_{ij} + p_{ji})/2$. L_G is a Laplacian matrix, and $L_G = D_G - \frac{P^T+P}{2}$.

We denote the *i-th* smallest eigenvalue of L_G as $\theta_i(L_P)$. Since L_P is semi-definitive, the eigenvalues are all non-negative. Therefore, the problem (15) can be equivalent to the following problem for a large enough value ofλ:

$$\begin{aligned} \min_{\forall i\ p_i^T 1=1,\ 0\le p_i\le 1} \sum_{i,j=1}^{m} \left(z_{ij}^d p_{ij} + \alpha p_{ij} + 2\gamma \sum_{i=1}^{e} \theta_i(L_P) \right) \\ s.t. \forall i,\ p_i^T 1 = 1,\ 0 \le p_i \le 1, \end{aligned} \tag{16}$$

where α stands for the regularization parameter.

Using Ky Fan's Theorem, we have

$$\sum_{i=1}^{e} \theta_i(L_P) = \min_{H\in R^{m\times c}, H^T H=I} Tr\left(H^T L_P H\right). \tag{17}$$

Thus, the problem (16) is further reformulated as the following equivalent form:

$$\begin{aligned} \min_{P,H} \sum_{i,j=1}^{m} \left(z_{ij}^d p_{ij} + \alpha p_{ij}^2 + 2\gamma Tr\left(H^T L_P H\right)\right) \\ s.t. \forall i,\ p_i^T 1 = 1,\ 0 \le p_{ij} \le 1,\ H \in R^{m\times c},\ H^T H = I. \end{aligned} \tag{18}$$

It can be found that the alternative optimization scheme can be used to solve the problem (18). By fixing P, the problem in Eq. (18) is rewritten as follows:

$$\min_{H\in R^{m\times c},\ H^T H=I} Tr\left(H^T L_P H\right). \tag{19}$$

The optimization solution H of Eq. (19) can be formed by the eigenvectors of L_P the corresponding c smallest eigenvalues.

By fixing H, we convert Eq. (19) into the following problem:

$$\begin{aligned} J_{opt} = \min_{G} \sum_{i,j=1}^{m} \left(z_{ij}^d p_{ij} + \alpha p_{ij}^2 \right) + 2\gamma Tr\left(H^T L_P H\right) \\ s.t.\ p_i^T 1 = 1, \forall j\ 0 \le p_{ij} \le 1, \end{aligned} \tag{20}$$

Then Eq. (20) can be simplified as the following form:

$$\begin{aligned} J_{opt} = \min_{p_i} \sum_{i,j=1}^{m} \left(z_{ij}^d p_{ij} + \alpha p_{ij}^2 + \gamma ||h_i - h_j||_2^2 p_{ij} \right) \\ s.t. p_i^T 1 = 1, \forall j, 0 \le p_{ij} \le 1. \end{aligned} \tag{21}$$

Denote $z_{ij}^h = ||h_i - h_j||_2^2$, the j-th element of z_j can be given as $z_{ij} = z_{ij}^x + \gamma z_{ij}^f$. Therefore, Eq. (21) is transformed as the following optimization problem:

$$\min_{p_i^T 1=1, 0\le p_{i\le 1}} ||p_i + \frac{1}{2\alpha} z_i||_2^2. \tag{22}$$

We construct the Lagrangian function for the problem (21) as follows:

$$L(p_i, \lambda, \eta_i) = \frac{1}{2}||p_i + \frac{z_i^d}{2\alpha_i}||_2^2 - \lambda\left(p_i^T 1 - 1\right) - \eta_i^T p_i, \tag{23}$$

Both λ and η are the non-negative Lagrangian multipliers. Using the Karush–Kuhn–Tucker (KKT) condition, the optimal solution p_i is derived as follows:

$$p_{ij} = \left(z_{ij}^x / 2\alpha_i + \lambda\right)_+. \tag{24}$$

where $(\bullet)_+$ denotes a non-negative value, and thus the value of p_{ij} is non-negative.

To reduce the computational cost, it seeks to get a sparse weight matrix P of the adaptive graph. Here, the number of neighbors of each point is set to k. Therefore, the value of α_i is set as follows:

$$\alpha_i = \frac{k}{2} z_{i,k+1}^x - \frac{1}{2}\sum_{j=1}^{k} z_{ij}^x. \tag{25}$$

Therefore, α is set to the average value of α_1 to α_k. Then we have:

$$\alpha = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{k}{2} z_{i,k+1}^x - \frac{1}{2}\sum_{j=1}^{k} z_{ij}^x\right). \tag{26}$$

Finally, the adaptive graph in the *l-th* layer is constructed as follows:

$$\left(w_{ij}\right) = p_{ij}. \tag{27}$$

3.3 Adaptive Graph Regularized Deep Semi-nonnegative Matrix Factorization

To fully consider the local geometric structure of the representation coefficient in each layer with parameter-free, the proposed AGRDSNMF method integrates the adaptive graph regularizer into the model of DSNMF in each layer. Therefore, the objective function of the proposed AGRDSNMF method in the l-th layer is given as follows:

$$\begin{gathered} O_l = ||X_l - U_l V_l||_F^2 + \omega Tr\left(V_l^T L_l V_l\right) \\ s.t.\ V_{ij} \geq 0, \end{gathered} \tag{28}$$

where ω denotes the regularization parameter. $L_l = D_l - W_l$ is a Laplacian matrix, where D_l is a diagonal matrix, and $(D_{ii})_l = \left(\sum_j W_{ij}\right)_l$. In Eq. (28), the first term is used to measure the approximation error, and the second term denotes the adaptive graph regularizer. However, Eq. (28) is a non-convex function for both U_l and V_l together, and thus we cannot achieve its global optimum. Fortunately, Eq. (28) becomes a convex problem while fixing one of the variables. In this way, we can achieve a local optimization minimum of the problem (28).

According to matrix properties, we have the following equations: $tr(XY) = tr(YX)$ and $tr(X) = tr\left(X^T\right)$. Thus, the problem (28) can be further rewritten as follows:

$$\begin{aligned} O_l &= tr\left((X_l - U_l V_l)(X_l - U_l V_l)^T\right) + \omega tr\left(V_l^T L_l V_l\right) \\ &= tr\left(X_l X_l^T\right) - 2tr\left(X_l V_l^T U_l\right) + tr\left(U_l V_l V_l^T U_l^T\right) + \omega tr\left(V_l^T L_l V_l\right). \end{aligned} \tag{29}$$

Let $(\varsigma_{ik})_l$ and $\left(\sigma_{jk}\right)_l$ be the Laplacian multipliers, and $\delta = \left[(\varsigma_{ik})_l\right]$, $\tau = \left[\left(\sigma_{jk}\right)_l\right]$. The Lagrange function in the *i-th* layer is defined as:

$$\begin{aligned}\kappa_l = tr\left(X_l X_l^T\right) - 2tr\left(X_l V_l^T U_l\right) + tr\left(U_l V_l V_l^T U_l^T\right) \\ +\omega tr\left(V_l^T L_l V_l\right) + tr\left(\delta U_l^T\right) + tr(\tau V_l).\end{aligned} \tag{30}$$

By taking partial derivatives of U_l and V_l, we can get:

$$\frac{\partial \kappa_l}{\partial U_l} = -X_l V_l^T + U_l V_l^T V_l + \omega L_l U_l + \delta, \tag{31}$$

$$\frac{\partial \kappa_l}{\partial V_l} = -2X_l^T U_l + 2V_l^T U_l^T U_l + 2\omega L_l V_l + \tau. \tag{32}$$

Using *KKT* conditions $(\varsigma_{ik} U_{ik})_l = 0$ and $\left(\sigma_{jk} V_{jk}\right)_l = 0$, we have

$$-\left(X_l U_l^T\right)_{ik} + \left(U_l V_l V_l^T\right)_{ik} W_{ik} = 0, \tag{33}$$

$$-\left(X_l^T U_l\right)_{jk} V_{jk}^T + \left(V_l^T U_l^T U_l\right)_{jk} + \omega\left(L_l V^T\right)_{jk} V_{jk}^T = 0. \tag{34}$$

(1) Updating rule for U_l.
By fixing V_l, the updating rule of U_l is given as follows:

$$\begin{aligned}U_l \leftarrow \left(Q^T Q\right)^{-1} Q^T X_l \tilde{V}_l^T \left(\tilde{V}_L \tilde{V}_l^T\right)^{-1} \\ U_l \leftarrow Q^{\dagger} X_l \tilde{V}_l^{\dagger}.\end{aligned} \tag{35}$$

(2) Updating rule for V_l while fixing U_l.
By fixing U_l, the updating rule of V_l is driven as follows:

$$V_l = V_l \odot \sqrt{\frac{\left[Q^T X_l\right]^+ + \left[Q^T Q\right]^- + \omega V_l W_l}{\left[Q^T X_l\right]^+ + \left[Q^T Q\right]^- + \omega V_l D_l}}. \tag{36}$$

where $Q = U_1 \ldots U_{l-1}$, and $\overset{\sim}{V_l}$ denotes the reconstruction of feature matrix in *l-th* layer. It is reconstructed as follows:

$$\overset{\sim}{V_l} = \begin{cases} V_l, \ l = t \\ U_{l+1} \underset{l+1}{\tilde{V}}, \ l \neq t \end{cases}, \tag{37}$$

where t is the number of the layer.
The procedure of the proposed AGRDSNMF algorithm is summarized as follows:

Algorithm 1: The algorithm of AGRDSNMF

Input: The nonnegative data matrix $X \in R^{m\times n}$, the regularized parameter ω, and the layer number t.

Initialize the basis matrix $U \in R^{m\times k}$ and the coefficient matrix $V \in R^{k\times m}$.
For l=1 to m do
Initialize U_l and V_l using the Semi-NMF method.
end for

While not converge **do**
For l=1 to m do
1. Generating adaptive graph
Step 1: update $L_{P_l} = D_{P_l} - (P_l^T + P_l)/2$.
Step 2: update H_l by using the eigenvectors of L_{P_l} the corresponding c smallest eigenvalues.
Step 3: For each l, update the l-th column of P_i by solving problem (13).
Step 4: $W_l = P_l$
2. Update the basis matrix U_l according to Eq. (35).
3. Update the coefficient matrix V_l according to Eq. (37).
end
Output: U and V.

4 Experiments and results

In this section, we conducted some experiments to evaluate the quality of the proposed method. Therefore, the proposed AGRDSNMF method is used to compare with other competitors including k-means, NMF, Semi-NMF, GNMF, DSNMF, and graph regularized Deep Semi-NMF (GDSNMF) on COIL20, PIE_pose27, TDT2 datasets. In order to measure the clustering performance, two classical evaluation metrics, such as accuracy (AC) and normalized mutual information (NMI), are used in real applications.

4.1 Evaluation metrics for Clustering

AC aims to measure the relationship of clusters and classes, and its definition is given as

$$\mathrm{AC} = \frac{\sum_{i=1}^{n} \delta(s_i, map(r_i))}{n}, \tag{38}$$

where s_i and r_i denote the ground-truth label, and the cluster label of the data point x_i, respectively. $map(\cdot)$ is a function mapping r_i to the equivalent label from the dataset. $\delta(x, y)$ denotes the delta function, and its definition is given as

$$\delta(x, y) = \begin{cases} 1, x = y \\ 0, x \neq y \end{cases}. \tag{39}$$

Let C and C' denote two sets of clusters, $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c_j' \in C'} p(c_i, c_j') \cdot \log_2 \frac{p(c_i, c_j')}{p(c_i)p(c_j')}, \tag{40}$$

where $p(c_i)$ and $p(c_j')$ are the probabilities of the sample belonging to the corresponding clusters, and $p(c_i, c_j')$ represents the joint probability that the selected sample belong to c_i and $c_j^{'}$ at the same time.

Therefore, NMI is defined as follows:

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

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively.

Purity is used to measures the extent to which clusters contain a single class. p_{ij} represents the probability of the members in cluster i belonging to class j. Thus, the definition of p_{ij} is given as follows:

$$p_{ij} = \frac{m_{ij}}{m_i} \tag{42}$$

where m_i denotes the sample number of i-th class, and m_{ij} denotes the number of samples in i-th cluster belonging to j-th class. Therefore, the purity can be defined as follows:

$$purity = \sum_{i=1}^{c} \frac{m_i}{m} p_i. \tag{43}$$

where c denotes the clustering number and $p_i = max(p_{ij})$.

4.2 Experiments on COIL20 Dataset

There are 1440 image samples from 20 objects on the COIL20 dataset. All the objects were taken every 5 degrees and thus each object includes 72 image samples. All samples were resized to 32×32 pixels with 256 grey levels per pixel. Part of sample images selected from the COIL20 dataset are shown in Fig. 1.

The first experiment was conducted on the COIL20 dataset. In each time, *some* samples were chosen at random to evaluate the effectiveness of our proposed algorithm. To make a fair comparison, the experiments were repeated five times, and their average results were computed as the final result for comparison The clustering results of different algorithms on the COIL20 dataset are summarized in Table 1.

It can be seen from this table, the AC of both GDSNMF and the proposed AGRDSNMF method achieve 5.0% and 7.7% improvement compared with DSNMF, respectively. The NMI of GDSNMF and AGRDSNMF achieve 5.6% and 12.6% improvement in contrast with DSNMF, respectively. That is because graph regularization in both GDNSMF and AGRDSNMF can make full use of geometric manifold structure between data. Besides, it can be seen that our proposed AGRDSNMF method outperforms GDSNMF regardless of the values of K. That is because a predefined fixing graph is adopted to model the geometric structure of data. However, it is not an optimal graph and thus the intrinsic structure of the complex data cannot be effectively explored. However, the proposed AGRDSNMF method can discover

Fig. 1 Sample images from the COIL20 dataset

the structure information embedded in data more effectively using an adaptive graph regularizer in each layer with parameter-free. Therefore, it is more conducive to apply different problems in practice.

4.3 Experiments on PIE_pose27 Dataset

There are 41,368 images from 68 subjects in total on the PIE dataset. Those volunteers were taken photos with different intensities of light, poses and facial expressions. Each image sample was stored in a 1024-dimensional vector. As a subset of PIE, PIE_pose27 contains 21 image samples for each individual under different lighting conditions. Part of images from the PIE_ pose27 dataset are shown in Fig. 2.

We investigated the superiority of the proposed AGRDSNMF method with the clustering number ranging from 30 to 68. Similarly, we run all methods for five times and reported their average values for comparison. The clustering performances of different algorithms on the PIE_pose27 dataset are shown in Table 2. It is easy to find that the performances of GNMF, GDSNMF and ARGDSNMF are better than NMF. This is because these three methods model the manifold structure of data with the graph regularization technology. Meanwhile, we can see that the proposed AGRDSNMF algorithm significantly outperforms GDSNMF. The main reason is that AGRDSNMF can learn an optimal graph model in each layer without additional parameters, while GDSNMF uses a predefined nearest neighbor graph. Therefore,

Table 1 The clustering results on the COIL20 dataset

Clusters	*k*-means	NMF	GNMF	SNMF	DSNMF	GDSNMF	AGRDSNMF
AC							
10	0.720	0.671	0.731	0.737	0.719	0.757	**0.771**
12	0.753	0.715	0.732	0.736	0.719	0.778	**0.819**
14	0.678	0.667	0.724	0.680	0.679	0.744	**0.767**
16	0.688	0.689	0.725	0.731	0.701	0.758	**0.773**
18	0.662	0.630	0.671	0.676	0.652	0.689	**0.728**
20	0.647	0.604	0.690	0.651	0.647	0.689	**0.718**
AVG	0.691	0.663	0.712	0.702	0.686	0.736	**0.763**
NMI							
10	0.760	0.716	0.848	0.748	0.716	0.802	**0.864**
12	0.798	0.763	0.852	0.765	0.755	0.829	**0.903**
14	0.779	0.750	0.810	0.759	0.834	0.813	**0.875**
16	0.784	0.768	0.859	0.761	0.777	0.830	**0.888**
18	0.760	0.736	0.829	0.707	0.730	0.801	**0.885**
20	0.767	0.731	0.827	0.742	0.731	0.805	**0.883**
AVG	0.775	0.744	0.838	0.747	0.757	0.813	**0.883**
Purity							
10	0.761	0.78	0.857	0.809	0.809	0.79	**0.904**
12	0.757	0.811	0.8	0.852	0.847	0.866	**0.876**
14	0.757	0.752	0.766	0.728	0.733	0.757	**0.826**
16	0.752	0.823	0.738	0.695	0.766	0.923	**0.928**
18	0.826	0.857	0.855	0.871	0.89	0.906	**0.912**
20	0.766	0.766	0.776	0.78	0.7428	0.914	**0.923**
AVG	0.770	0.798	0.799	0.789	0.798	0.859	**0.895**

Fig. 2 Some sample images from the PIE_pose27 dataset

Table 2 The clustering results on the PIE_pose27 dataset

Clusters	*k*-means	NMF	GNMF	SNMF	DSNMF	GDSNMF	AGRDSNMF
AC							
30	0.266	0.332	0.579	0.351	0.394	0.560	**0.710**
35	0.258	0.312	0.533	0.334	0.342	0.608	**0.674**
40	0.264	0.311	0.494	0.322	0.361	0.514	**0.671**
45	0.246	0.294	0.495	0.282	0.349	0.515	**0.658**
50	0.252	0.182	0.492	0.287	0.264	0.477	**0.642**
55	0.247	0.178	0.471	0.270	0.245	0.445	**0.603**
60	0.240	0.160	0.493	0.250	0.200	0.431	**0.624**
65	0.236	0.156	0.502	0.253	0.196	0.443	**0.567**
68	0.245	0.145	0.476	0.247	0.195	0.415	**0.573**
AVG	0.287	0.230	0.504	0.288	0.283	0.490	**0.636**
NMI							
30	0.481	0.418	0.791	0.547	0.748	0.811	**0.854**
35	0.497	0.412	0.780	0.543	0.466	0.782	**0.857**
40	0.522	0.426	0.747	0.547	0.503	0.738	**0.856**
45	0.521	0.414	0.749	0.520	0.499	0.746	**0.847**
50	0.530	0.415	0.766	0.544	0.514	0.741	**0.842**
55	0.532	0.418	0.764	0.538	0.509	0.722	**0.830**
60	0.534	0.401	0.772	0.522	0.474	0.721	**0.823**
65	0.537	0.402	0.785	0.535	0.471	0.724	**0.821**
68	0.547	0.398	0.779	0.529	0.483	0.704	**0.812**
AVG	0.522	0.412	0.770	0.536	0.519	0.743	**0.838**
Purity							
30	0.375	0.457	0.667	0.587	0.452	0.751	**0.801**
35	0.349	0.447	0.56	0.547	0.542	0.74	**0.759**
40	0.285	0.449	0.706	0.566	0.529	**0.722**	0.637
45	0.396	0.417	0.706	0.608	0.523	0.804	**0.817**
50	0.349	0.436	0.671	0.69	0.642	0.764	**0.82**
55	0.33	0.349	0.653	0.645	0.452	0.733	**0.793**
60	0.33	0.417	0.719	0.632	0.476	0.862	**0.873**
65	0.298	0.505	0.656	0.587	0.611	0.825	**0.851**
68	0.354	0.402	0.589	0.608	0.537	0.668	**0.738**
AVG	0.336	0.431	0.659	0.608	0.529	0.763	**0.788**

the representation coefficient of AGRDSNMF can exploit the intrinsic geometry structure among data effectively.

4.4 Experiments on TDT2 Dataset

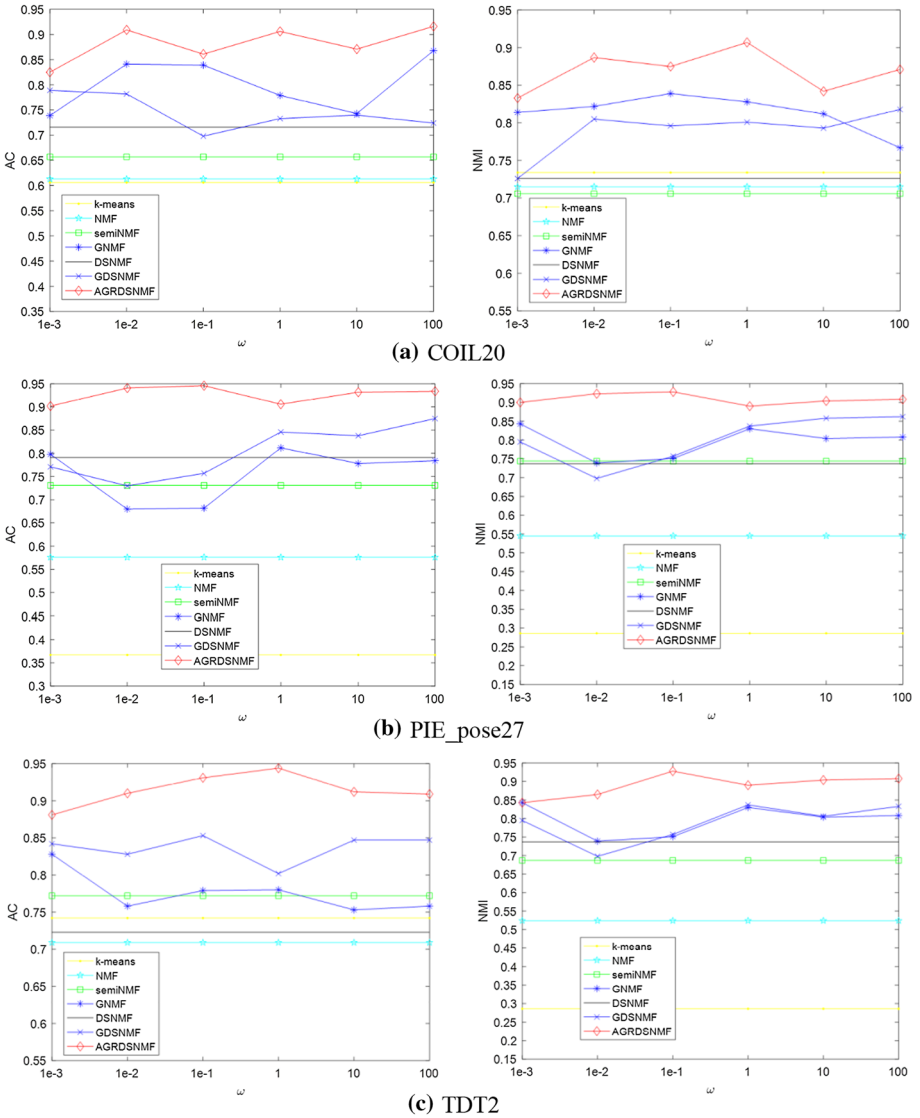
TDT2 dataset includes 11,201 on-top documents, and all of them are divided into 96 semantic types. All data were taken from those six sources:2 television programs (ABC, CNN), 2 radio

programs (PRI, VOA), and 2 newswires (NYT, APW). TDT2 is a subset of TDT2_all by removing documents that appear in two or more categories and preserving the largest 30 categories 9,394 documents in total.

Similarly, *some* samples were randomly chosen as a subset. All methods were carried out five times and their average values were reported. The clustering results of all methods on the TDT2 dataset are summarized in Table 3. It can be seen that the average performance of SNMF is inferior to both DSNMF and GDSNMF. Specifically, the clustering AC of DSNMF and GDSNMF is 31.0% and 34.2% higher than SNMF, respectively. And the clustering NMI of DSNMF and GDSNMF are 32.6% and 35.7% higher than SNMF respectively. It can be found that GDSNMF outperforms DSNMF regardless of AC and NMI. This is because

Table 3 The clustering results on the TDT2 dataset

Clusters	*k*-means	NMF	GNMF	SNMF	DSNMF	GDSNMF	AGRDSNMF
AC							
16	0.704	0.422	0.749	0.466	0.652	0.773	**0.797**
18	0.733	0.420	0.488	0.429	0.341	0.727	**0.784**
20	0.717	0.339	0.668	0.393	0.485	**0.765**	0.732
22	0.750	0.354	0.591	0.412	0.477	0.760	**0.802**
24	0.744	0.628	0.638	0.409	0.511	0.771	**0.792**
26	0.649	0.332	0.581	0.385	0.355	0.701	**0.787**
28	0.745	0.320	0.655	0.352	0.371	0.699	**0.776**
30	0.697	0.309	0.629	0.369	0.280	**0.755**	0.730
AVG	0.717	0.391	0.624	0.402	0.434	0.744	**0.775**
NMI							
16	0.732	0.523	0.713	0.391	0.525	0.800	**0.837**
18	0.702	0.517	0.640	0.416	0.502	0.806	**0.815**
20	0.731	0.454	0.765	0.410	0.464	0.803	**0.832**
22	0.718	0.485	0.708	0.439	0.595	0.797	**0.824**
24	0.800	0.456	0.741	0.455	0.506	0.807	**0.822**
26	0.778	0.461	0.710	0.430	0.363	0.780	**0.806**
28	0.766	0.469	0.753	0.480	0.396	0.784	**0.803**
30	0.784	0.470	0.763	0.495	0.415	0.802	**0.811**
AVG	0.752	0.479	0.724	0.440	0.471	0.797	**0.819**
Purity							
16	0.771	0.847	0.811	0.855	0.523	0.86	**0.896**
18	0.761	0.742	0.728	0.761	0.566	0.753	**0.800**
20	0.668	0.828	0.838	0.752	0.533	0.892	**0.919**
22	0.676	0.766	0.795	0.707	0.695	0.883	**0.904**
24	0.711	0.819	0.761	0.723	0.652	0.784	**0.800**
26	0.742	0.757	0.757	0.633	0.376	**0.776**	0.747
28	0.799	0.811	0.7	0.836	0.623	0.852	**0.947**
30	0.79	0.79	0.779	0.862	0.823	0.805	**0.890**
AVG	0.740	0.795	0.771	0.766	0.599	0.826	**0.863**



(a) COIL20

(b) PIE_pose27

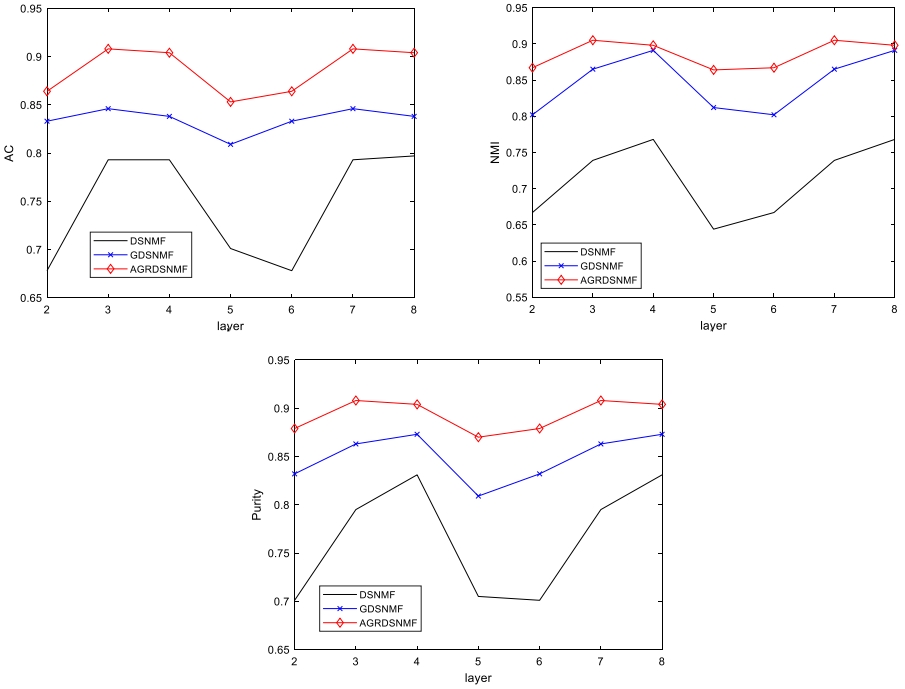
(c) TDT2

Fig. 3 The performances varied with the vales of the parameter ω

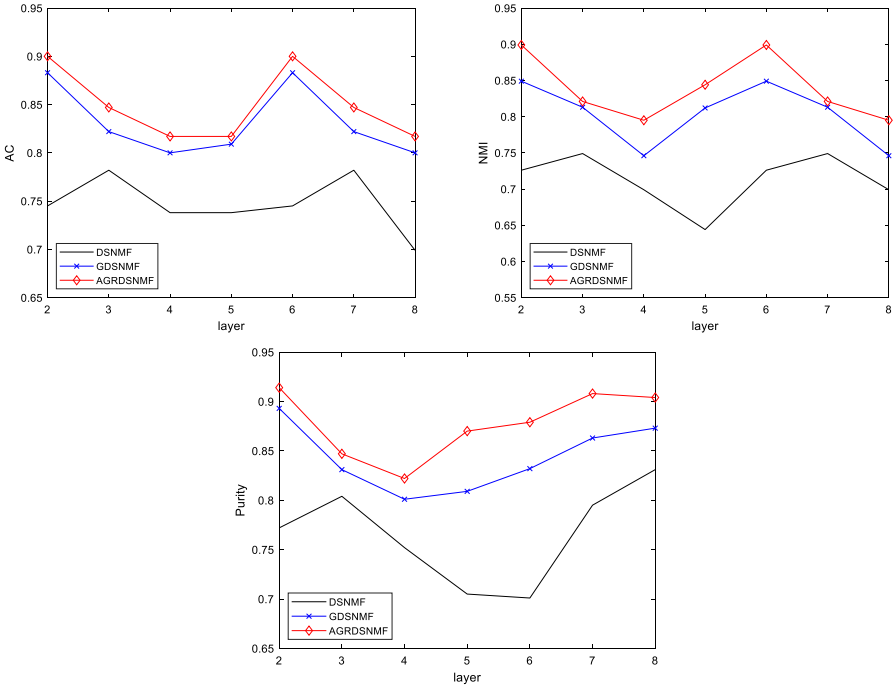
GDSNMF considers the manifold structure of data by constructing a fixing graph. In addition, we can see that our AGRDSNMF method shows superiority to other compared methods. That is because the proposed AGRDSNMF method can effectively explore the intrinsic geometry structure information among data by constructing an optimal adaptive graph model.

4.5 Parameter Sensitivity

In this subsection, the experiments were carried out to evaluate the sensitivity of the regularization parameter. The proposed AGRDSNMF method includes the only one regularized

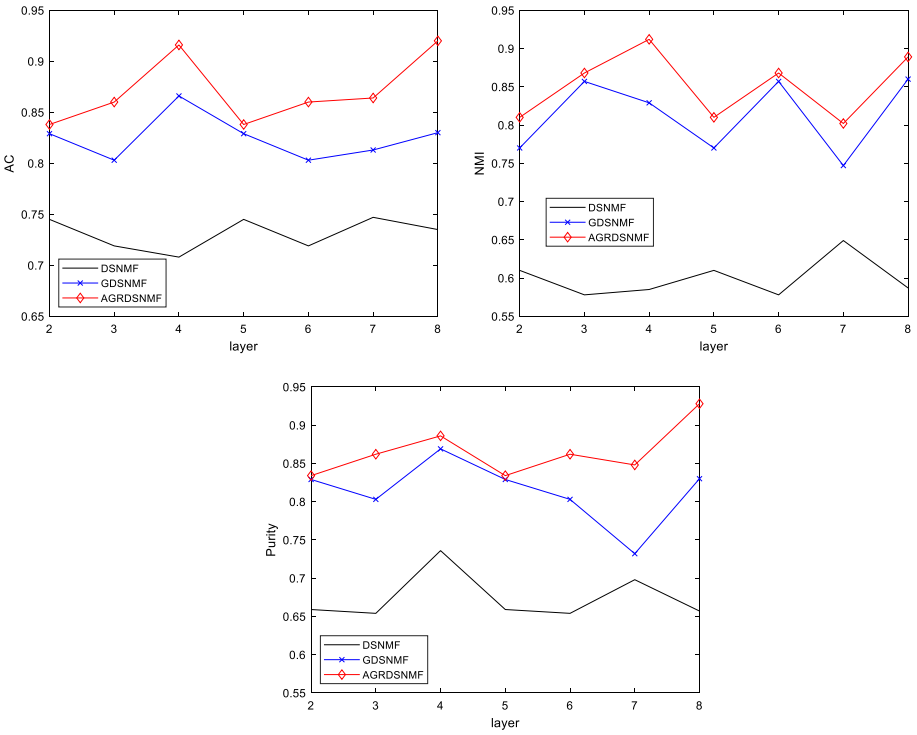


(a) COIL20



(b) PIE_pose27

Fig. 4 The performances under different layers



(c) TDT2

Fig. 4 continued

parameter ω. In particular, the parameter ω is turned among the setting {0.001, 0.01, 0.1, 1, 10, 100}. Similarly, we randomly selected the category samples from three datasets. The clustering performances under different settings of the parameter ω of different methods on COIL20, PIE_ pose27, and TDT2 datasets are shown in Fig. 3. It is easy to find that our AGRDSNMF method shows stable results regardless of the parameter ω changing greatly.

4.6 Experiments of Layers

In this subsection, we experimentally study the influence of the deep matrix factorization algorithms, such as DSNMF、GDSNMF and AGRDSNMF, under different setting of the layers. 18, 60, 26 categories samples were chosen randomly from COIL20, PIE_pose27 and TDT datasets to conduct the clustering experiments, respectively. Figure 4 shows the clustering results of three methods based on deep architecture varied with the setting of layers on three benchmark datasets. We can see that AGRDSNMF achieves better clustering performance than both DSNMF and GDSNMF regardless of the number of layers on three datasets.

5 Conclusions

In this work, an adaptive graph regularized deep semi-nonnegative matrix factorization (AGRDSNMF) is proposed for data clustering. Similar to the traditional deep NMF algorithms, the proposed AGRDSNMF also adopts a deep framework to explore the hidden feature information of data. Besides, AGRDSNMF can handle the data by mixing the noise since both data matrix and basis matrix are allowed to have negative components. Therefore, the proposed AGRDSNMF method can be widely applied to different real problems. Moreover, our proposed AGRDSNMF method effectively captures the intrinsic local relationship between samples by learning an optimal adaptive graph in each layer, rather than the predefined fixed graph. And then the adaptive graph regularizer is integrated into the matrix factorization framework. In addition, a solution based on the multiplicative iteration algorithm is developed to efficiently optimize the proposed objective function. Extensive experiments conducted on real datasets have validated that our proposed AGRDSNMF method other state-of-the-art competitors in clustering.

Acknowledgements This work was supported by the National Natural Science Foundation of China [Grant No. 61603159, 62162033, U21B2027, 61902160], Yunnan Provincial Major Science and Technology Special Plan Projects [Grant No. 202002AD080001, 202103AA080015], Yunnan Foundation Research Projects [Grant No. 202101AT070438, 202101BE070001-056], Excellent Key Teachers of QingLan Project in Jiangsu Province.

References

1. Ma J, Zhang Y, Zhang L, Du B, Tao D (2019) Pseudo supervised matrix factorization in discriminative subspace. In: International Joint Conference on Artificial Intelligence, (IJCAI), pp 4554–4560
2. Nie F, Pei S, Wang R, Li X (2020) Fast clustering with co-clustering via discrete non-negative matrix factorization for image identification. In: IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp 2073–2077
3. Wang D, Gao X, Wang X, He L (2019) Label consistent matrix factorization hashing for large-scale cross-modal similarity search. IEEE Trans Pattern Anal Mach Intell 41(10):2466–2479
4. Zhang D, Wu X-J (2020) Scalable discrete matrix factorization and semantic autoencoder for cross-media retrieval. IEEE Trans Cybern. https://doi.org/10.1109/TCYB.2020.3032017
5. Wang D, Wang Q, Gao X (2018) Robust and flexible discrete hashing for cross-modal similarity search. IEEE Trans Circuits Syst Video Technol 28(10):2703–2715
6. Du B, Wang S, Wang N et al (2016) Hyperspectral signal unmixing based on constrained non-negative matrix factorization approach. Neurocomputing 204:153–161
7. Wang N, Du B, Zhang L (2013) An endmember dissimilarity constrained non-negative matrix factorization method for hyperspectral unmixing. IEEE J Sel Top Appl Earth Observ Remote Sens 6(2):554–569
8. Cai D, He X, Han J (2011) Locally consistent concept factorization for document clustering. IEEE Trans Knowl Data Eng 23(6):902–913
9. Salehani YE, Arabnejad E, Rahiche A et al (2020) MSdB-NMF: MultiSpectral document image binarization framework via non-negative matrix factorization approach. IEEE Trans Image Process 29:9099–9112
10. Jiao C, Gao Y, Yu N et al (2020) Hyper-graph regularized constrained NMF for selecting differentially expressed genes and tumor classification. IEEE J Biomed Health Inform 24(10):3002–3011
11. Wang C, Yu N, Wu M et al (2021) Dual hyper-graph regularized supervised NMF for selecting differentially expressed genes and tumor classification. IEEE/ACM Trans Comput Biol Bioinf 18(6):2375–2383
12. Jia Y, Liu H, Hou J, Kwong S (2021) Semisupervised adaptive symmetric non-negative matrix factorization. IEEE Trans Cybern 51(5):2550–2562
13. Meng Y, Shang R, Shang F et al (2020) Semi-supervised graph regularized deep NMF with Bi-orthogonal constraints for data representation. IEEE Trans Neural Networks Learn Syst 31(9):3245–3258
14. Lee D, Seung H et al (2001) Algorithms for non-negative matrix factorization. Adv Neural Inf Process Syst 13:556–562
15. Ding C, Li T, Jordan M (2010) Convex and semi-nonnegative matrix factorizations. IEEE Trans Softw Eng 32(1):45–55

16. Xu W, Gong Y (2004) Document clustering by concept factorization. In: Proceedings of the 27th annual international conference on Research and development in information retrieval, vol 202, pp 202–209
17. Cai D, He X, Han J et al (2011) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 33(8):1548–1560
18. Shang F, Jiao LC, Wang F (2012) Graph dual regularization non-negative matrix factorization for co-clustering. Pattern Recogn 45:2237–2250
19. Shu Z, Wu X, You C et al (2020) Rank-constrained nonnegative matrix factorization for data representation. Inf Sci 528:133–146
20. Ghassemian R et al (2015) Spectral unmixing of hyperspectral imagery using multilayer NMF. IEEE Geosci Remote Sens Lett 12(1):38–42
21. Shu Z, Zhou J, Tong L, Bai X, Zhao C et al (2015) Multilayer manifold and sparsity constrained nonnegative matrix factorization for Hyperspectral Unmixing. In: IEEE International Conference on Image Processing (ICIP), pp 1–8
22. Tong L, Zhou J, Qian B et al (2019) Adaptive graph regularized multilayer nonnegative matrix factorization for Hyperspectral Unmixing. IEEE J Sel Top Appl Earth Observ Remote Sens 13:434–447
23. Fang H, Li A, Xu H, Wang T (2018) Sparsity-constrained deep nonnegative matrix factorization for Hyperspectral Unmixing. IEEE Geosci Remote Sens Lett 15(7):1105–1109
24. Feng X, Li H, Li J et al (2018) Hyperspectral Unmixing using sparsity-constrained deep nonnegative matrix factorization with Total Variation. IEEE Trans Geosci Remote Sens 56(10):6245–6257
25. Zhao Y, Wang H, Pei J (2019) Deep non-negative matrix factorization architecture based on underlying basis images learning. In: IEEE Transactions on Pattern Analysis and Machine Intelligence, Early Access Article
26. Trigeorgis G, Bousmalis K, Zafeiriou S et al (2015) A deep matrix factorization method for learning attribute representations. IEEE Trans Pattern Anal Mach Intell 39(3):417–429
27. Schmidhuber J (2015) Deep learning in neural networks: an overview. Neural Networks 61:85–117
28. Krizhevsky A, Sutskever I, Hinton G (2012) ImageNet classification with deep convolutional neural networks. In: Advances in Neural Information Processing Systems
29. He W, Zhang H, Zhang L (2017) Total variation regularized reweighted sparse nonnegative matrix factorization for hyperspectral unmixing. IEEE Trans Geosci Remote Sens 55(7):3909–3921
30. Leng C, Cai G, Yu D, Wang Z (2017) Adaptive total-variation for non-negative matrix factorization on manifold. Pattern Recognit Lett 98:68–74
31. Shu Z, Wu X, Fan H et al (2017) Parameter-less auto-weighted multiple graph regularized nonnegative matrix factorization for data representation. Knowl-Based Syst 131:105–112
32. Nie F, Wang X, Huang H (2014) Clustering and projected clustering with adaptive neighbors. In: The 20th ACM SIGKDD Conference on Knowledge Discovery and Data Mining (KDD), pp 997–986