Egyptian Informatics Journal 22 (2021) 91–100

Contents lists available at ScienceDirect

Egyptian Informatics Journal

journal homepage: www.sciencedirect.com

Full length article

Joint non-negative and fuzzy coding with graph regularization for efficient data clustering

Yong Peng [a,b,*], Yikai Zhang [a], Feiwei Qin [a], Wanzeng Kong [a,c]

[a] School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China
[b] Provincial Key Laboratory for Computer Information Processing Technology, Soochow University, Suzhou 215123, China
[c] Key Laboratory of Brain Machine Collaborative Intelligence of Zhejiang Province, Hangzhou 310018, China

ARTICLE INFO

Article history:
Received 11 February 2020
Revised 4 May 2020
Accepted 10 May 2020
Available online 30 May 2020

Keywords:
Non-negative matrix factorization
Fuzzy coding
Local coordinate coding
Graph regularization
Clustering

ABSTRACT

Non-negative matrix factorization (NMF) is an effective model in converting data into non-negative coefficient representation whose discriminative ability is usually enhanced to be used for diverse pattern recognition tasks. In NMF-based clustering, we often need to perform *K*-means on the learned coefficient as postprocessing step to get the final cluster assignments. This breaks the connection between the feature learning and recognition stages. In this paper, we propose to learn the non-negative coefficient matrix based on which we jointly perform fuzzy clustering, by viewing that each column of the dictionary matrix as a concept of each cluster. As a result, we formulate a new fuzzy clustering model, termed Joint Non-negative and Fuzzy Coding with Graph regularization (G-JNFC), and design an effective optimization method to solve it under the alternating direction optimization framework. Besides the convergence and computational complexity analysis on G-JNFC, we conduct extensive experiments on both synthetic and representative benchmark data sets. The results show that the proposed G-JNFC model is effective in data clustering.

1. Introduction

In the community of machine learning, the basic pipeline is first transforming data into its feature representation based on which we can perform pattern recognition tasks. As an effective learning model, the NMF [1,2] can transform the original data into the corresponding coefficient representation which can be viewed as the feature representation of data with usually enhanced discriminative ability. Then diverse pattern recognition tasks can be conducted on the learned coefficient matrix such as clustering [3,4], semi-supervised learning [5,6], classification [7,8] and some specific applications [9,10]. Generally, most of the improved NMF variants achieved acceptable performance in diverse applications.

In the field of data clustering, after obtaining the coefficient representation matrix in NMF, we often need to perform *K*-means to get the final cluster assignment of each sample. This suggests that the coefficient representation matrix itself cannot act as the cluster indicator matrix to directly provide us the clustering results. Therefore, such two-stage paradigm, 'feature extraction plus recognition', becomes the usual way to NMF-based recognition tasks and this paradigm is shared by many other models such as sparse subspace clustering [11–13], and low rank representation based subspace clustering [14–16]. However, there exist at least two limitations in this paradigm. One is that it inevitably breaks the connection between these two steps and causes extra computational burden in the postprocessing step. The other is that a promising clustering method should not only determine the membership of each data point to a specified cluster, but also provide us more detailed information towards the samples in the boundary areas of different clusters. Therefore, fuzzy clustering [17,18] which can depict the confidence of each data point to different clusters sometimes more coincides with our intuitive understanding to the knowledge of data when compared with hard clustering.

To overcome the above mentioned two limitations in conventional matrix factorization based clustering, we propose an improved model to make the learned coefficient matrix simultaneously achieve the final clustering results and provide the membership degree of each data point to different clusters. The newly formulated model is termed as Joint Non-negative and Fuzzy Coding with Graph regularization (G-JNFC). In G-JNFC, apart from the

* Corresponding author at: 1158 No.2 Avenue, Xiasha Higher Education Zone, Jianggan District, Hangzhou 310018, China.
E-mail address: yongpeng@hdu.edu.cn (Y. Peng).

Peer review under responsibility of Faculty of Computers and Information, Cairo University.

https://doi.org/10.1016/j.eij.2020.05.001

graph regularization and local coordinate coding [19], we additionally constrain the summation of row elements in the coefficient representation matrix to be one. This allows the proposed G-JNFC model to be illustrated from two different perspectives, *improved local coordinate coding* and *joint representation learning and fuzzy clustering*. We propose an efficient algorithm to optimize the objective function of G-JNFC. Once G-JNFC is solved, we can categorize each data point to a specified cluster by checking the largest value of each row of the coefficient representation matrix. Moreover, if there are multiple non-zero values in each row, this means the corresponding data point has fuzzy membership to different clusters. The computational complexity and convergence of the optimization method are given. Extensive experiments on both synthetic and benchmark data sets show the effectiveness of G-JNFC.

The remainder of this paper is organized as follows. We give a brief introduction to the related works, NMF and local coordinate coding, in Section 2. The proposed G-JNFC model including the model formulation, optimization, convergence and complexity analysis are given in Section 3. In Section 4, we conduct extensive experiments to evaluate the effectiveness of G-JNFC. Section 5 concludes the whole paper.

2. Related works

The NMF was proposed to learn parts-based representation by mathematically approximating the data matrix by two non-negative factor matrices as

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_2^2, s.t. \mathbf{U}\geqslant \mathbf{0}, \mathbf{V}\geqslant \mathbf{0}, \tag{1}$$

where $\mathbf{X}\in\mathbb{R}^{d\times n}$ is the data matrix, $\mathbf{U}\in\mathbb{R}^{d\times c}$ is the basis matrix in which each column is expected to capture the essence of data points in one cluster, $\mathbf{V}\in\mathbb{R}^{n\times c}$ is the coefficient matrix. Here, d, n, c are the dimensionality, number of data points and number of classes (clusters), respectively. This objective in (1) can be optimized by alternately updating the two variables. Then, $\mathbf{V}$ is usually employed as feature in pattern recognition tasks.

If we view the basis matrix $\mathbf{U}$ as a set of anchor points and enforce each data point to be represented by a linear combination of only a few nearby anchor points, the non-negative local coordinate factorization (NLCF) [20] was formulated by incorporating the following constraint [19] into the NMF objective as a regularization term

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n}\sum_{k=1}^{c} v_{ik}\|\mathbf{x}_i-\mathbf{u}_k\|_2^2. \tag{2}$$

To be specific, if $\mathbf{x}_i$ is closer to $\mathbf{u}_k$, we will have a larger v_{ik}. Otherwise, we will have a smaller v_{ik} when $\mathbf{x}_i$ and $\mathbf{u}_k$ are dissimilar.

In clustering tasks, both NMF and NLCF rely on performing K-means on the learned $\mathbf{V}$ to obtain the final results.

3. The proposed model

In this section, we make an improvement on the existing matrix factorization model to let it learn direct and interpretable clustering results.

3.1. Model formulation

We formulate the objective function of the joint non-negative and fuzzy coding (JNFC) model as

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_2^2+\lambda\sum_{i=1}^{n}\sum_{k=1}^{c} v_{ik}\|\mathbf{x}_i-\mathbf{u}_k\|_2^2,\\ &s.t. \mathbf{U}\geqslant \mathbf{0}, \mathbf{V}\geqslant \mathbf{0}, \mathbf{V}\mathbf{1}=\mathbf{1}.\end{aligned} \tag{3}$$

Obviously, we can find that JNFC objective additionally enforces the summation of each row of the coefficient matrix to one. Though it is a minor modification, we declare that 1) JNFC can directly achieve the clustering results by checking the largest value of each row of the coefficient matrix without performing post-processing; and 2) JNFC can provide us the membership degree of each data point to different clusters, leading to a fuzzy clustering model.

There are two different perspectives to depict the motivation of the second term in (3) which are respectively explained below.

- *Improved local coordinate coding* [21]. Mathematically, we can view NMF as linearly representing each data point by columns of the dictionary matrix. From this point of view, it is obvious that the corresponding representation coefficient v_{ik} should be larger if the data point $\mathbf{x}_i$ is closer to a specific column $\mathbf{u}_k$. If we use the squared Euclidean distance to measure the closeness, one reasonable objective to reach this fact is

$$\min_{\mathbf{U},\mathbf{V}} \sum_{i=1}^{n}\sum_{k=1}^{c} v_{ik}\|\mathbf{x}_i-\mathbf{u}_k\|_2^2. \tag{4}$$

- *Joint representation learning and fuzzy clustering*. To group data into each cluster with a certain membership degree, the objective of fuzzy K-means clustering can be defined as

$$\min_{\mathbf{Y}\mathbf{1}=\mathbf{1},\mathbf{Y}\geqslant\mathbf{0},\boldsymbol{\mu}_j} \sum_{i=1}^{n}\sum_{j=1}^{c} y_{ij}\|\mathbf{x}_i-\boldsymbol{\mu}_j\|_2^2, \tag{5}$$

where c is the number of clusters and $\mathbf{Y}\in\mathbb{R}^{n\times c}$ is the fuzzy membership. Moreover, $\boldsymbol{\mu}_j$ is the centroid of the j-th cluster and the clustering centroid matrix is $\mathbf{C}\triangleq[\boldsymbol{\mu}_1,\boldsymbol{\mu}_2,\cdots,\boldsymbol{\mu}_c]\in\mathbb{R}^{d\times c}$. Each element y_{ij} of matrix $\mathbf{Y}$ represents the membership degree of the i-th sample belonging to the j-th cluster [22]. By following the convention that the number of columns of the basis matrix is set as the number of clusters, each column of the basis matrix may be called a 'concept' to characterize the essential property of a cluster. In other words, each column of the basis matrix should be approximately equivalent to the centroid of a certain cluster. Based on this analysis, we can view the second term in (3) as a modified fuzzy K-means objective. Then, the proposed JNFC model can be seen as the combination of NMF and fuzzy K-means.

In addition, we also consider the local consistency of the learned representation coefficient matrix by incorporating the graph regularizer [23] into the JNFC model and then formulate the Joint Non-negative and Fuzzy Coding with Graph regularization (G-JNFC) model as

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_2^2 \quad +\lambda\sum_{i=1}^{n}\sum_{k=1}^{c} v_{ik}\|\mathbf{x}_i-\mathbf{u}_k\|_2^2+\gamma\mathrm{Tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}),\\ &s.t. \mathbf{U}\geqslant \mathbf{0}, \mathbf{V}\geqslant \mathbf{0}, \mathbf{V}\mathbf{1}=\mathbf{1}.\end{aligned} \tag{6}$$

Obviously, G-JNFC will degenerate to JNFC when the second regularization parameter γ approaches zero. In general, our proposed G-JNFC have the following three main advantages.

- First, G-JNFC is a unified model for joint data representation learning and clustering. In most of existing studies, the learned coefficient matrix is treated as the data representation on which K-means is additionally performed to get the final cluster assignments. In G-JNFC, once the coefficient matrix is obtained, we can obtain the final cluster assignment of each data point by checking the largest value of each row of the coefficient matrix.
- Second, G-JNFC takes the data manifold into consideration along which the learned coefficient is constrained to vary smoothly. This local invariance idea is of great significance in improving the performance of learning models.

- Third, G-JNFC takes the connection between the basis matrix and the coefficient matrix into consideration. That is, G-JNFC uses a binary constraint relating both variables in the model, which can be seen as a second-order constraint.

3.2. Optimization method

Below we give an elegant optimization method to the objective function of G-JNFC in (6). Generally, it is under the alternating direction optimization framework in which we put the emphasis on the derivation of updating rule to $\mathbf{V}$.

■ Update $\mathbf{U}$ with $\mathbf{V}$ fixed. We can rewrite the objective function of (6) as

$$\min_{\mathbf{U}\geqslant\mathbf{0}}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_2^2+\lambda\sum_{i=1}^{n}\|(\mathbf{x}_i\mathbf{1}^T-\mathbf{U})\mathbf{\Lambda}_i^{1/2}\|_2^2, \tag{7}$$

where $\mathbf{\Lambda}_i=diag(\mathbf{v}_i^T)\in\mathbb{R}^{c\times c}$ is a diagonal matrix and $\mathbf{v}_i^T$ is the i-th row of representation coefficient matrix $\mathbf{V}$. The corresponding Lagrangian function is

$$\mathcal{L}(\mathbf{U})=\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_2^2+\lambda\sum_{i=1}^{n}\|(\mathbf{x}_i\mathbf{1}^T-\mathbf{U})\mathbf{\Lambda}_i^{1/2}\|_2^2+\mathrm{Tr}(\mathbf{\Phi}\mathbf{U}^T).$$

Taking the derivative of $\mathcal{L}$ with respect to $\mathbf{U}$, we have

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}}=2\mathbf{U}\mathbf{V}^T\mathbf{V}-2\mathbf{X}\mathbf{V}+\lambda\sum_{i=1}^{n}(-2\mathbf{x}_i\mathbf{1}^T\mathbf{\Lambda}_i+2\mathbf{U}\mathbf{\Lambda}_i)+\mathbf{\Phi}.$$

Using KKT condition $\phi_{jk}u_{jk}=0$ $(j=1,\cdots,d,k=1,\cdots,c)$, we get the following equation

$$\begin{aligned}(\mathbf{U}\mathbf{V}^T\mathbf{V})_{jk}u_{jk}\quad&-(\mathbf{X}\mathbf{V})_{jk}u_{jk}+\lambda\Big(\sum_{i=1}^{n}\mathbf{U}\mathbf{\Lambda}_i\Big)_{jk}u_{jk}\\&-\lambda\Big(\sum_{i=1}^{n}\mathbf{x}_i\mathbf{1}^T\mathbf{\Lambda}_i\Big)_{jk}u_{jk}=0\end{aligned} \tag{8}$$

This leads to the updating rule of $\mathbf{U}$ as

$$u_{jk}\leftarrow u_{jk}\frac{\Big(\mathbf{X}\mathbf{V}+\lambda\sum_{i=1}^{n}\mathbf{x}_i\mathbf{1}^T\mathbf{\Lambda}_i\Big)_{jk}}{\Big(\mathbf{U}\mathbf{V}^T\mathbf{V}+\lambda\sum_{i=1}^{n}\mathbf{U}\mathbf{\Lambda}_i\Big)_{jk}}, \tag{9}$$

which can be rewritten in compact matrix form as

$$u_{jk}\leftarrow u_{jk}\frac{((\lambda+1)\mathbf{X}\mathbf{V})_{jk}}{(\mathbf{U}\mathbf{V}^T\mathbf{V}+\lambda\mathbf{U}\mathbf{H})_{jk}}. \tag{10}$$

Here $\mathbf{H}\in\mathbb{R}^{c\times c}$ is a diagonal matrix whose elements are column sums of $\mathbf{V}$.

■ Update $\mathbf{V}$ with $\mathbf{U}$ fixed. Denoting $e_{ik}=\|\mathbf{x}_i-\mathbf{u}_k\|^2$ as the (i,k)-th element of matrix $\mathbf{E}\in\mathbb{R}^{n\times c}$, we have

$$\begin{aligned}&\min_{\mathbf{V}\geqslant\mathbf{0},\mathbf{V}\mathbf{1}=\mathbf{1}}\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|_2^2+\lambda\mathrm{Tr}(\mathbf{E}\mathbf{V}^T)\\\Longleftrightarrow\quad&\min_{\mathbf{V}\geqslant\mathbf{0},\mathbf{V}\mathbf{1}=\mathbf{1}}\mathrm{Tr}(\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T)-\mathrm{Tr}(\mathbf{V}(2\mathbf{U}^T\mathbf{X}-\lambda\mathbf{E}^T))\end{aligned} \tag{11}$$

In this paper, we denote the i-th row of $\mathbf{V}$ as $\mathbf{v}_i^T$, and $\mathbf{v}_i$ is a column vector whose elements are corresponding to those of $\mathbf{v}_i^T$. Then the third term in (6) can be rewritten as

$$\min_{\mathbf{v}_i\geqslant\mathbf{0},\mathbf{v}_i^T\mathbf{1}=1}\sum_{i=1}^{n}\sum_{j=1}^{n}\|\mathbf{v}_i-\mathbf{v}_j\|_2^2s_{ij}, \tag{12}$$

where s_{ij} is the (i,j)-th element of the graph affinity matrix $\mathbf{S}$. The relation between the graph Laplacian $\mathbf{L}$ and $\mathbf{S}$ is $\mathbf{L}=\mathbf{D}-\mathbf{S}$, where $\mathbf{D}\in\mathbb{R}^{n\times n}$ is a diagonal degree matrix with the i-th diagonal element as $d_{ii}=\sum_{i=1}^{n}s_{ij}$. Therefore, we propose to row-wisely optimize $\mathbf{V}$ in (6), and the objective function associated with $\mathbf{v}_i$ is

$$\min_{\mathbf{v}_i}\mathbf{v}_i^T\mathbf{A}\mathbf{v}_i-\mathbf{v}_i^T\mathbf{b},\ s.t.\mathbf{v}_i\geqslant\mathbf{0},\mathbf{v}_i^T\mathbf{1}=1, \tag{13}$$

where $\mathbf{A}\triangleq\mathbf{U}^T\mathbf{U}+\gamma d_{ii}\mathbf{I}\in\mathbb{R}^{c\times c},\mathbf{b}=(\mathbf{U}^T\mathbf{X}-\lambda\mathbf{E}^T)_i+2\gamma\sum_{j=1}^{n}s_{ij}\mathbf{v}_j\in\mathbb{R}^{c\times1}$ and $(\mathbf{U}^T\mathbf{X}-\lambda\mathbf{E}^T)_i$ is the i-th column of matrix $\mathbf{U}^T\mathbf{X}-\lambda\mathbf{E}^T$.

To solve (13), we introduce an auxiliary variable $\mathbf{z}$ (and simultaneously remove the subscript to simplify the expression) to rewrite it as

$$\min_{\mathbf{v}\geqslant\mathbf{0},\mathbf{v}^T\mathbf{1}=1,\mathbf{z}}\mathbf{z}^T\mathbf{A}\mathbf{v}-\mathbf{z}^T\mathbf{b}. \tag{14}$$

Based on the augmented Lagrangian multiplier (ALM) method, problem (14) can be rewritten as

$$\min_{\mathbf{v}\geqslant\mathbf{0},\mathbf{v}^T\mathbf{1}=1,\mathbf{z}}\mathbf{z}^T\mathbf{A}\mathbf{v}-\mathbf{z}^T\mathbf{b}+\frac{\mu}{2}\left\|\mathbf{v}-\mathbf{z}+\frac{\boldsymbol{\beta}}{\mu}\right\|_2^2, \tag{15}$$

Accordingly, an alternative optimization method is applied to solve problem (15).

1) Fixing $\mathbf{v}$, *update* $\mathbf{z}$. By taking the derivative of problem (15) with respect to $\mathbf{z}$ and setting it to zero, we have

$$\mathbf{z}=\mathbf{v}+\frac{\mathbf{b}+\boldsymbol{\beta}-\mathbf{A}\mathbf{v}}{\mu}. \tag{16}$$

2) Fixing $\mathbf{z}$, *update* $\mathbf{v}$.When $\mathbf{z}$ fixed, problem (15) becomes

$$\min_{\mathbf{v}\geqslant\mathbf{0},\mathbf{v}^T\mathbf{1}=1}\mathbf{z}^T\mathbf{A}\mathbf{v}+\frac{\mu}{2}\left\|\mathbf{v}-\mathbf{z}+\frac{\boldsymbol{\beta}}{\mu}\right\|_2^2, \tag{17}$$

which can be reformulated to the following form by completing the squared form of $\mathbf{v}$

$$\min_{\mathbf{v}\geqslant\mathbf{0},\mathbf{v}^T\mathbf{1}=1}\frac{\mu}{2}\left\|\mathbf{v}-\mathbf{z}+\frac{1}{\mu}(\boldsymbol{\beta}+\mathbf{A}^T\mathbf{z})\right\|_2^2. \tag{18}$$

Subproblem (18) is a Euclidean projection on the simplex [24–26]. Here we present an effective algorithm to solve (18).

Denote $\mathbf{m}=\mathbf{z}-\frac{1}{\mu}(\boldsymbol{\beta}+\mathbf{A}^T\mathbf{z})$ and then (18) is equivalent to

$$\min_{\mathbf{v}\geqslant\mathbf{0},\mathbf{v}^T\mathbf{1}=1}\frac{1}{2}\|\mathbf{v}-\mathbf{m}\|_2^2. \tag{19}$$

The Lagrangian function of (19) is

$$\mathcal{L}(\mathbf{v},\delta,\boldsymbol{\eta})=\frac{1}{2}\|\mathbf{v}-\mathbf{m}\|_2^2-\delta(\mathbf{v}^T\mathbf{1}-1)-\boldsymbol{\eta}^T\mathbf{v}, \tag{20}$$

where $\delta,\boldsymbol{\eta}$ are Lagrangian multipliers to be determined. If $\mathbf{v}^*$ is the optimal solution, and $\delta^*,\boldsymbol{\eta}^*$ are the corresponding multipliers, based on the KKT condition, we have the following equations for each $t\in[1,c]$

$$\begin{cases}v_t^*-m_t-\delta^*-\eta_t^*=0,\\v_t^*\geqslant0,\\\eta_t^*\geqslant0,\\v_t^*\eta_t^*=0\end{cases} \tag{21}$$

The first equation in (21) can be rewritten as

$$\mathbf{v}^*-\mathbf{m}-\delta^*\mathbf{1}-\boldsymbol{\eta}^*=\mathbf{0}. \tag{22}$$

According to the constraint $\mathbf{v}^T\mathbf{1}=1$, the above equation can be reformulated into

$$\delta^*=\frac{1-\mathbf{1}^T\mathbf{m}-\mathbf{1}^T\boldsymbol{\eta}^*}{c}. \tag{23}$$

By substituting (23) into (22), we have

$$\mathbf{v}^*=\mathbf{m}-\frac{\mathbf{1}\mathbf{1}^T}{c}\mathbf{m}+\frac{1}{c}\mathbf{1}-\frac{\mathbf{1}^T\boldsymbol{\eta}^*}{c}\mathbf{1}+\boldsymbol{\eta}^*. \tag{24}$$

Denote $\bar{\eta}^* = \frac{\mathbf{1}^T\boldsymbol{\eta}^*}{c}$ and $\mathbf{g} = \mathbf{m} - \frac{\mathbf{1}\mathbf{1}^T}{c}\mathbf{m} + \frac{1}{c}\mathbf{1}$, we have

$$\mathbf{v}^* = \mathbf{g} + \boldsymbol{\eta}^* - \bar{\eta}^*\mathbf{1}. \tag{25}$$

Hence, for each $t \in [1, c]$, we have

$$v_t^* = g_t + \eta_t^* - \bar{\eta}^*. \tag{26}$$

According to (21) and (26), we have $g_t + \eta_t^* - \bar{\eta}^* = (g_t - \bar{\eta}^*)_+$, where $(a)_+ = \max(a, 0)$. Therefore, we have

$$v_t^* = (g_t - \bar{\eta}^*)_+. \tag{27}$$

It is obvious that if $\bar{\eta}^*$ can be determined, the optimal solution to $\mathbf{v}^*$ can be obtained from the above equation. In a similar way, we can rewrite (26) as $\eta_t^* = v_t^* + \bar{\eta}^* - g_t$ such that $\eta_t^* = (\bar{\eta}^* - g_t)_+$. Therefore, $\bar{\eta}^*$ can be calculated by

$$\bar{\eta}^* = \frac{1}{c}\sum_{t=1}^{c}(\bar{\eta}^* - g_t)_+. \tag{28}$$

According to $\mathbf{v}^{*T}\mathbf{1} = 1$ and (27), we can define a function as

$$f(\bar{\eta}) = \sum_{t=1}^{c}(g_t - \bar{\eta})_+ - 1. \tag{29}$$

When the above function equals to zero, the optimal $\bar{\eta}^*$ can be obtained by Newton method [27] to find the root of (29) as

$$\bar{\eta}_{k+1} = \bar{\eta}_k - \frac{f(\bar{\eta}_k)}{f\prime(\bar{\eta}_k)}. \tag{30}$$

Consequently, the procedure of row-wisely optimizing $\mathbf{V}$ in objective (13) is given in Algorithm 2 and the whole procedure of our proposed G-JNFC model is summarized in Algorithm 3.

3.3. Analysis on convergence and complexity

On the convergence of the proposed G-JNFC model, here we give the high level analysis on its optimization procedure. Since the optimization of $\mathbf{U}$ is identical to that in NLCF, we can construct the same auxiliary function, *i.e.*, Eq. (14), as shown in the appendix of [20] to prove that the objective function of G-JNFC is nonincreasing under the updating rule of (10). When updating the variable $\mathbf{V}$, objective (13) is a quadratic programming problem with constraints [28]. Since the matrix $\mathbf{A}$ is symmetric and positive semi-definite, the optimal solution could be obtained. Then, Algorithm 2 converges to the global optimal solution. As a whole, we conclude that the convergence to the optimization of G-JNFC can be guaranteed. In the section of experiments, we will show the monotonically decreasing of objective function values in terms of the iterations.

The complexity analysis of the optimization procedure of G-JNFC objective function is based on the big $\mathcal{O}$ notation and the usual case of $n > d \gg c$.

- *Updating matrix* $\mathbf{U}$. The complexity of updating $\mathbf{U}$ by (10) is caused by the matrix multiplication operation, leading to $\mathcal{O}(dnc)$ complexity.
- *Updating matrix* $\mathbf{V}$. When updating each row of the variable $\mathbf{V}$, the main complexity is $\mathcal{O}(c^2)$ which comes from the calculation of $\mathbf{v}_i$ in (13), leading to $\mathcal{O}(nc^2)$ complexity in updating $\mathbf{V}$.
- *Updating matrix* $\mathbf{A}$ *and vector* $\mathbf{b}$. When calculating the intermediate matrix $\mathbf{A}$ and vector $\mathbf{b}$, the multiplication of $\mathbf{U}^T\mathbf{U}$ and $\mathbf{U}^T\mathbf{X}$ with complexities $\mathcal{O}(dc^2)$ and $\mathcal{O}(dnc)$ can be ignored since it can be calculated out of the loop of Algorithm 2.

Assuming that the iteration numbers of Algorithms 1 and 2 are t_1 and t_2, respectively, we have the overall complexity of the proposed G-JNFC model is $\mathcal{O}(t_1(dnc + t_2nc^2))$. From the following experiments, we will find that the optimization converges in a few iterations.

Algorithm 1 The algorithm to solve subproblem (19).

Input: the given vector $\mathbf{m} \in \mathbb{R}^c$;
Output: the target vector $\mathbf{v} \in \mathbb{R}^c$.
1: Calculate $\mathbf{g} = \mathbf{m} - \frac{\mathbf{1}\mathbf{1}^T}{c}\mathbf{m} + \frac{1}{c}\mathbf{1}$;
2: Obtain the root $\bar{\eta}^*$ of (29) based on Newton's method;
3: Get the optimal solution $v_t^* = (g_t - \bar{\eta}^*)_+$ for each $t \in [1, c]$;

Algorithm 2: The algorithm to solve subproblem (13)

Input: matrix $\mathbf{A} \in \mathbb{R}^{c\times c}$ and vector $\mathbf{b} \in \mathbb{R}^c$;
Output: vector $\mathbf{v} \in \mathbb{R}^c$.
1: Initialize $\mathbf{v}, \mu, \boldsymbol{\beta}$, and ρ;
2: while not converged do
3: Update $\mathbf{z}$ by (16);
4: Update $\mathbf{v}$ by solving problem (18) via Algorithm 1;
5: Update $\boldsymbol{\beta} = \boldsymbol{\beta} + \mu(\mathbf{v} - \mathbf{z})$;
6: Update $\mu = \rho\mu$;
7: end while

Algorithm 3 The overall procedure of G-JNFC.

Input: the data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$ (d and n represent the feature vector length and the data size, respectively), the number of clusters c, regularization parameters λ and γ;
Output: The clustering result.
1: Calculate the graph affinity matrix $\mathbf{S} \in \mathbb{R}^{n\times n}$ based on the 'HeatKernel' function with the neighboring size 5 and the bandwidth parameter as the mean of the squared pairwise distances;
2: Randomly initialize $\mathbf{V}$ to satisfy $\mathbf{V} \geqslant \mathbf{0}, \mathbf{V}\mathbf{1} = \mathbf{1}$;
3: // The outer loop is to optimize the objective (6) of G-JNFC
4: while not converged do
5: // Update the basis matrix $\mathbf{U}$
6: Update $\mathbf{U}$ based on (10);
7: // The inner loop is to optimize the coefficient matrix $\mathbf{V}$ rowwisely
8: for $i = 1 : n$
9: Prepare matrix $\mathbf{A}$ and vector $\mathbf{b}$ w.r.t. the i-th row of $\mathbf{V}$;
10: Update $\mathbf{v}_i$ by optimizing the objective (13) based on Algorithm 2;
11: end for
12: end while
13: Obtain the clustering results by checking the largest value of each row of $\mathbf{V}$.

4. Experiments

In this section, experiments are conducted on both synthetic and representative benchmark data sets to evaluate the performance of our proposed G-JNFC model.

4.1. Experiment on synthetic data

To intuitively show the fuzzy effect of G-JNFC in data clustering, we generate a synthetic data set composed of five Gaussian distributed clusters, as shown in the left rand side of Fig. 1. The blue cluster has 200 data points and each of the remaining clusters has 75 data points. We visualize the clustering results obtained by G-JNFC in the right hand side of Fig. 1 in which the numbers of the data points with fuzzy membership towards different clusters are highlighted. The learned coefficient matrix, which simultaneously acts as the fuzzy clustering indicator matrix, is shown in Fig. 2. In Fig. 2, the abscissa values represent the 500 data points and ordinate values of the five subplots show the membership degrees of data points to the blue, red, green, cyan and purple clusters, respectively. We can observe the obvious correspondence between the numbered data points in Fig. 1 and their non-zero membership degrees towards different clusters in Fig. 2.

Below we give the explanations to some representative numbered samples of two different scenarios. 1) The first one is that though the data point was generated by one cluster, it is surrounded by data points from the other cluster. For example, the 33rd data point was originally generated by the blue Gaussian distribution; however, it is obviously much closer to the data points from the purple cluster based on our visualization. Therefore, it is more reasonable to group it into the purple cluster. The 214th and 433rd data points share similar properties. 2) The other scenario is that if a data point is in the boundary area of multiple clusters, it may have non-zero membership degrees to those clusters. For example, the learned coefficient vector corresponding to the 70th data point, *i.e.*, $\mathbf{v}_{70}^T$, is [0.721,0,0.279,0,0], meaning that it is in the boundary area of the first (blue) cluster and the third (green) cluster. However, it is more reasonable to group it into the blue cluster if we have to determine its only one membership (0.721>0.279). From Fig. 1, we can find that the 70th data point is actually more closer to the blue cluster, reflecting the consistency between the visualization and the computational result. There are also some similar data points such as the 110th, 140th, and 145th ones.

Based on the above experimental results on synthetic data, we can conclude that the fuzzy information depicted by G-JNFC is more consistent to our intuitive understanding to the knowledge involved in the data.

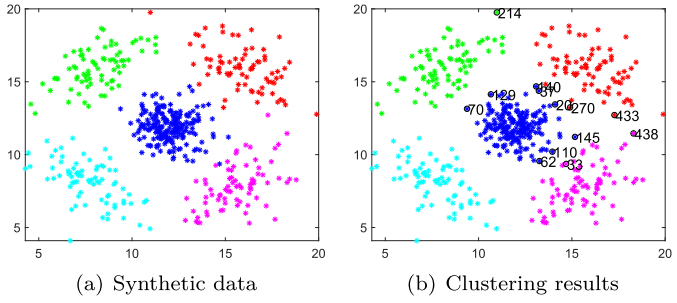


Fig. 1. Synthetic Gaussian distributed data (left) and the clustering results obtained by G-JNFC (right) which are best viewed in color.

4.2. Experiment on benchmark data

4.2.1. Evaluation metrics

To evaluate the clustering results, we employ the three widely used metrics, *i.e.*, *Accuracy* (ACC), *Normalized Mutual Information* (NMI) and *Purity*, to measure the clustering performance [29] whose definitions are given below.

Accuracy exploits the one-to-one relationship between clusters and classes. It measures the extent to which each cluster contained samples from the corresponding class, which has the following definition

$$\mathrm{Acc} = \frac{1}{n}\sum_{i=1}^{n}\delta(\mathrm{map}(r_i), l_i), \tag{31}$$

where r_i denotes the cluster label of $\mathbf{x}_i$ and l_i denotes its ground truth label, n is the sample size, $\delta(x,y)$ equals to one if $x = y$ and zero otherwise. $\mathrm{map}(r_i)$ is the permutation mapping function which can map each cluster label r_i to the equivalent class label.

Normalized Mutual Information (NMI) is used to determine the quality of clusters. Given a clustering result, the NMI is estimated by

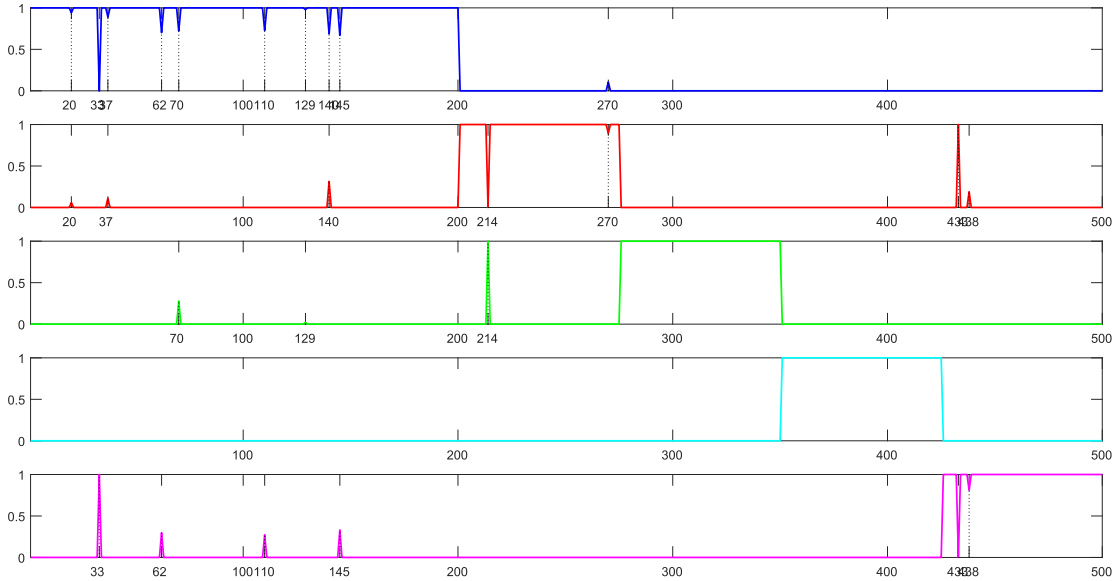


Fig. 2. The membership degrees of each data point to different clusters reflected by the learned coefficient matrix **V** of G-JNFC.

$$\mathrm{NMI} = \frac{\sum_{i=1}^{c}\sum_{j=1}^{c} n_{ij}\log\frac{n_{ij}}{n_i\hat{n}_j}}{\sqrt{(\sum_{i=1}^{c} n_i\log\frac{n_i}{n})(\sum_{j=1}^{c}\hat{n}_j\log\frac{\hat{n}_j}{n})}} \tag{32}$$

where n_i represents the number of samples in the cluster $C_i(1 \leqslant i \leqslant c)$, $\hat{n}_j$ is the number of samples belonging to the class $L_j(1 \leqslant i \leqslant c)$, and n_{ij} is the number of samples in the intersection between cluster C_i and L_j.

Purity measures the extent to which each cluster primarily contained samples from one class. The clustering purity is the weighted sum of individual cluster purity values, given by

$$\mathrm{Purity} = \sum_{i=1}^{c}\frac{n_i}{n}P(S_i),\ \ P(S_i) = \frac{1}{n_i}\max_j(n_i^j), \tag{33}$$

where S_i is a certain cluster with size n_i, n_i^j is the number of the i-th input class which was assigned to the j-th cluster.

4.2.2. Data sets and experimental settings

Five representative benchmark data sets were used in the experiment including the **Caltech101** natural image data set, **TDT2** text data set, **dig1-10** digit image data set, **Yale** face image data set and the **Isolet** spoken letter data set. Their basic properties including the numbers of samples, feature dimensions and clusters were summarized in Table 1.

Table 1
Basic properties of the five data sets used in the experiments.

Data Set	# Samples	# Dimension	# Cluster
Caltech101	3044	500	10
TDT2	1938	36771	20
dig1-10	1797	64	10
Yale	165	1024	15
Isolet	1560	617	26

In order to show the effectiveness of our G-JNFC model, we compare it with some state-of-the-art clustering methods including *K*-means, Non-negative Matrix Factorization (NMF) [30], concept factorization (CF) [31], normalized cut (NCut), fuzzy-*C*-means (FCM), nonnegative local coordinate factorization (NLCF) and its graph regularized version [20]. Moreover, we include the comparison of JNFC, which is equivalent to setting the regularization parameter γ to zero in G-JNFC. For graph-based clustering models including the NCut, GNLCF and G-JNFC, we set the number of nearest neighbors to 5 as suggested in [23]. The 'Heatkernel' function was used to measure the similarity of data pairs where the bandwidth parameter was set as the average value of pairwise distances.

If there is(are) free regularization parameter(s), we tuned it (them) from $\left\{10^{-5}, 10^{-4}, \cdots, 10^{5}\right\}$ to let the related methods achieve the best performance. The fuzzy order parameter in FCM was selected from $\{1, 1.1, \cdots, 2\}$. For methods which rely on *K*-means as postprocessing method to get the final results including NMF, CF, NCut, NLCF and GNLCF, we repeated *K*-means five times with different initializations and recorded the best results. For fuzzy clustering methods including FCM, JNFC and G-JNFC, we investigated the largest membership degree corresponding to the data point to determine its assignment.

4.2.3. Experimental results and analysis

Tables 2–6 show the clustering results on the Caltech101, TDT2, dig1-10 and Isolet data sets, respectively. In order to randomize the experiments, we conduct the evaluations with different cluster

Table 2
Clustering performance (%) on of compared algorithms Caltech101 data set.

c	*K*means	NMF	CF	NCut	NLCF	GNLCF	FCM	JNFC	G-JNFC
					Accuracy				
2	67.9 ± 23.1(1)	68.0 ± 10.2(1)	70.3 ± 14.2(1)	74.8 ± 16.3(1)	74.7 ± 10.4(1)	84.8 ± 8.5(0)	73.5 ± 11.7(1)	77.4 ± 11.6(1)	**86.9 ± 7.6**
3	60.5 ± 19.9(1)	59.4 ± 14.3(1)	61.2 ± 14.0(1)	66.2 ± 16.2(1)	66.5 ± 15.4(1)	72.6 ± 12.6(1)	66.6 ± 13.4(1)	68.1 ± 12.8(1)	**77.0 ± 12.5**
4	56.9 ± 17.5(1)	54.6 ± 10.8(1)	59.8 ± 12.9(1)	64.2 ± 11.8(0)	60.4 ± 12.8(1)	**69.0 ± 9.0**(0)	65.6 ± 12.6(0)	68.0 ± 11.0(0)	68.0 ± 7.3
5	57.9 ± 14.3(1)	54.8 ± 9.7(1)	51.7 ± 10.6(1)	57.6 ± 10.0(1)	57.3 ± 10.0(1)	65.0 ± 6.4(1)	63.5 ± 9.9(1)	65.3 ± 9.7(1)	**68.3 ± 8.8**
6	51.1 ± 10.3(1)	50.0 ± 7.1(1)	46.7 ± 8.3(1)	49.9 ± 7.0(1)	52.6 ± 7.6(1)	56.4 ± 3.8(1)	55.4 ± 6.7(1)	57.8 ± 6.3(1)	**60.2 ± 5.3**
7	49.7 ± 9.7(1)	45.2 ± 8.5(1)	43.4 ± 7.6(1)	52.8 ± 8.1(1)	46.4 ± 7.7(1)	53.7 ± 5.0(1)	53.0 ± 7.8(1)	59.3 ± 7.2(1)	**60.7 ± 6.6**
8	48.4 ± 6.9(1)	46.6 ± 3.4(1)	42.6 ± 6.3(1)	51.5 ± 4.9(1)	48.1 ± 3.7(1)	50.5 ± 4.5(1)	54.1 ± 4.7(1)	58.1 ± 5.0(1)	**59.4 ± 4.2**
9	45.1 ± 2.7(1)	43.9 ± 2.4(1)	39.0 ± 4.3(1)	46.3 ± 3.4(1)	45.2 ± 2.9(1)	45.5 ± 2.3(1)	49.4 ± 3.9(1)	56.5 ± 3.7(1)	**57.7 ± 2.8**
10	46.9	42.8	31.5	43.5	41.9	44.1	46.5	53.9	**55.1**
Avg.	53.8	51.7	49.6	56.3	54.8	60.2	58.6	62.7	**65.9**
					Normalized Mutual Information				
2	12.2 ± 23.1(1)	11.5 ± 21.9(1)	19.8 ± 24.1(1)	28.5 ± 31.7(0)	15.7 ± 22.3(1)	23.3 ± 31.2(1)	20.4 ± 24.4(1)	26.2 ± 28.1(0)	**33.9 ± 31.3**
3	23.0 ± 19.9(1)	21.4 ± 19.3(1)	27.4 ± 19.1(1)	33.6 ± 22.9(1)	31.6 ± 23.0(1)	30.1 ± 19.8(1)	30.4 ± 19.8(1)	34.2 ± 20.4(1)	**40.0 ± 21.8**
4	27.3 ± 17.5(1)	23.9 ± 15.3(1)	35.1 ± 13.6(1)	38.35 ± 12.4(0)	33.3 ± 16.8(1)	34.8 ± 17.9(1)	38.6 ± 14.9(0)	40.0 ± 13.6(0)	**44.3 ± 8.5**
5	35.4 ± 14.3(1)	31.4 ± 13.6(1)	34.0 ± 12.4(1)	36.72 ± 12.2(0)	36.4 ± 14.1(1)	26.2 ± 15.7(1)	40.4 ± 11.5(0)	**42.0 ± 10.3**(0)	41.0 ± 12.1
6	30.0 ± 10.3(0)	28.7 ± 9.7(0)	31.1 ± 7.8(0)	33.57 ± 7.8(0)	28.4 ± 13.5(0)	24.0 ± 14.3(1)	34.5 ± 8.8(0)	**36.1 ± 7.82**(1)	32.9 ± 10.1
7	35.1 ± 9.7(1)	30.0 ± 10.0(1)	33.3 ± 8.2(1)	37.79 ± 9.1(0)	33.2 ± 9.9(1)	24.4 ± 14.3(1)	36.2 ± 7.4(1)	**39.4 ± 8.6**(0)	39.1 ± 9.0
8	36.4 ± 6.9(1)	35.2 ± 3.5(1)	34.4 ± 4.6(1)	38.77 ± 4.3(0)	35.9 ± 6.8(1)	26.5 ± 13.9(1)	35.2 ± 7.5(1)	**40.2 ± 4.8**(0)	39.8 ± 6.0
9	35.8 ± 2.7(1)	33.7 ± 3.9(1)	32.3 ± 3.1(1)	36.51 ± 3.0(1)	36.2 ± 2.5(1)	31.6 ± 11.1(1)	33.0 ± 7.7(1)	39.1 ± 2.9(0)	**39.6 ± 2.5**
10	36.7	32.4	28.8	35.8	34.3	37.1	37.7	37.8	**38.1**
Avg.	30.2	27.6	30.7	35.5	31.7	28.7	34.0	37.2	**38.8**
					Purity				
2	79.2 ± 9.4(1)	79.1 ± 9.4(1)	82.0 ± 7.8(1)	84.7 ± 9.9(1)	81.8 ± 7.8(1)	84.8 ± 8.5(1)	80.3 ± 9.4(1)	83.6 ± 8.8(1)	**87.4 ± 7.4**
3	67.9 ± 16.1(1)	67.2 ± 16.3(1)	72.0 ± 12.6(1)	74.7 ± 14.0(1)	71.3 ± 14.9(1)	73.7 ± 12.7(1)	69.6 ± 12.8(0)	75.1 ± 12.6(1)	**78.6 ± 12.6**
4	65.8 ± 12.2(0)	63.4 ± 9.3(1)	70.9 ± 8.2(0)	72.2 ± 9.1(0)	68.2 ± 11.1(0)	72.0 ± 9.4(0)	68.9 ± 11.5(0)	**74.0 ± 9.0**(1)	68.9 ± 7.8
5	69.6 ± 9.1(1)	66.8 ± 8.1(1)	70.9 ± 8.1(1)	71.7 ± 9.1(0)	71.0 ± 8.1(1)	67.5 ± 8.4(1)	70.4 ± 9.6(0)	**74.7 ± 7.5**(0)	73.9 ± 7.9
6	64.0 ± 5.8(1)	63.4 ± 5.6(1)	66.9 ± 4.8(0)	67.8 ± 5.1(0)	63.6 ± 6.8(1)	63.1 ± 7.9(1)	65.6 ± 4.2(0)	**70.0 ± 3.9**(1)	68.1 ± 5.4
7	63.6 ± 7.1(1)	59.2 ± 7.4(1)	64.2 ± 7.2(1)	67.6 ± 7.9(0)	62.2 ± 6.7(1)	59.9 ± 8.3(1)	59.1 ± 10.2(1)	67.8 ± 6.5(0)	**67.9 ± 6.6**
8	61.6 ± 5.6(1)	60.1 ± 3.2(1)	63.3 ± 4.0(1)	65.9 ± 3.7(0)	62.0 ± 4.6(1)	58.2 ± 9.3(1)	62.5 ± 6.4(1)	**67.0 ± 3.4**(0)	66.8 ± 4.1
9	60.0 ± 2.6(1)	57.8 ± 1.5(1)	60.1 ± 2.2(1)	63.0 ± 2.3(1)	60.3 ± 2.5(1)	59.8 ± 8.8(1)	61.9 ± 8.9(1)	64.9 ± 2.4(1)	**65.9 ± 2.0**
10	60.0	55.7	54.5	59.9	57.5	64.2	60.8	63.3	**64.7**
Avg.	65.7	63.6	67.2	69.7	66.4	67.0	66.6	71.2	**71.4**

Table 3
Clustering performance (%) of compared algorithms on TDT2 data set.

c	*K*means	NMF	CF	NCut	NLCF	GNLCF	FCM	JNFC	G-JNFC
					Accuracy				
2	99.8 ± 0.3(0)	99.8 ± 0.3(0)	99.9 ± 0.3(0)	99.7 ± 0.6(1)	99.9 ± 0.3(0)	99.9 ± 0.3(0)	99.8 ± 0.3(0)	99.8 ± 0.3(0)	**99.9 ± 0.2**
4	99.2 ± 0.9(1)	98.9 ± 1.7(1)	93.1 ± 11.9(1)	95.9 ± 8.6(0)	97.2 ± 7.0(0)	97.9 ± 3.1(1)	99.3 ± 0.9(1)	99.6 ± 0.5(1)	**99.8 ± 0.3**
6	94.7 ± 7.7(1)	94.3 ± 7.6(1)	86.9 ± 9.6(1)	93.2 ± 7.6(1)	95.5 ± 7.1(1)	90.3 ± 9.2(1)	97.7 ± 4.1(0)	98.7 ± 2.2(0)	**99.4 ± 0.4**
8	92.5 ± 7.0(1)	85.5 ± 9.3(1)	82.5 ± 12.6(1)	85.3 ± 5.3(1)	91.8 ± 6.8(1)	89.5 ± 6.4(1)	95.5 ± 7.9(1)	98.3 ± 3.0(1)	**99.3 ± 1.3**
10	91.4 ± 6.3(1)	85.7 ± 8.0(1)	85.7 ± 8.3(1)	82.7 ± 6.6(1)	90.0 ± 6.4(1)	85.8 ± 6.3(1)	91.3 ± 6.4(1)	96.7 ± 3.9(1)	**99.0 ± 0.6**
12	87.7 ± 4.7(1)	80.2 ± 5.9(1)	75.6 ± 7.4(1)	79.8 ± 6.3(1)	86.8 ± 5.6(1)	80.9 ± 5.6(1)	88.5 ± 4.4(1)	94.7 ± 2.9(1)	**97.1 ± 2.6**
14	86.8 ± 4.6(1)	83.0 ± 5.6(1)	77.8 ± 6.5(1)	78.7 ± 4.6(1)	83.5 ± 5.5(1)	78.7 ± 5.2(1)	87.8 ± 4.6(1)	94.0 ± 3.0(0)	**95.0 ± 3.1**
16	88.0 ± 3.7(1)	79.6 ± 6.8(1)	75.0 ± 6.2(1)	75.7 ± 4.2(1)	82.8 ± 4.6(1)	78.0 ± 4.8(1)	85.8 ± 4.1(1)	93.4 ± 2.5(0)	**93.6 ± 2.6**
18	85.5 ± 4.4(1)	73.6 ± 5.8(1)	73.1 ± 3.9(1)	77.7 ± 5.6(1)	82.4 ± 4.5(1)	78.0 ± 4.8(1)	84.4 ± 4.5(1)	91.4 ± 1.8(0)	**92.4 ± 2.1**
20	85.8	80.2	80.1	71.0	79.9	73.8	86.1	90.5	**91.5**
Avg.	91.1	86.1	83.0	84.0	89.0	85.3	91.6	95.7	**96.7**
					Normalized Mutual Information				
2	98.4 ± 2.6(1)	95.5 ± 2.8(0)	98.7 ± 2.5(0)	97.5 ± 4.4(1)	98.7 ± 2.7(0)	98.6 ± 2.5(0)	98.6 ± 2.5(0)	98.4 ± 2.6(1)	**99.5 ± 1.5**
4	97.3 ± 2.9(1)	96.7 ± 4.0(1)	92.5 ± 8.5(1)	94.8 ± 7.1(1)	95.3 ± 5.7(1)	94.4 ± 6.6(1)	97.5 ± 2.8(1)	98.2 ± 2.0(1)	**99.0 ± 1.3**
6	94.2 ± 5.5(1)	93.6 ± 6.2(1)	90.0 ± 5.7(1)	93.4 ± 5.3(1)	94.1 ± 6.2(1)	89.9 ± 7.0(1)	96.0 ± 3.9(1)	97.5 ± 2.2(0)	**98.3 ± 1.1**
8	94.0 ± 4.7(1)	89.1 ± 6.2(1)	87.9 ± 7.3(1)	88.6 ± 3.0(1)	91.1 ± 6.5(1)	89.2 ± 5.1(1)	95.4 ± 5.3(1)	97.1 ± 3.5(1)	**98.3 ± 2.5**
10	92.8 ± 4.0(1)	88.4 ± 6.4(1)	88.2 ± 5.3(1)	87.4 ± 4.6(1)	91.1 ± 5.3(1)	88.5 ± 4.6(1)	93.0 ± 4.1(1)	96.0 ± 3.3(1)	**97.8 ± 1.2**
12	90.3 ± 3.4(1)	86.5 ± 3.6(1)	83.2 ± 4.7(1)	85.1 ± 4.4(1)	89.2 ± 3.6(1)	85.9 ± 4.6(1)	91.1 ± 3.4(1)	94.3 ± 2.0(1)	**96.0 ± 2.0**
14	90.2 ± 3.3(1)	88.0 ± 2.8(1)	84.5 ± 3.5(1)	84.7 ± 3.5(1)	88.0 ± 3.6(1)	84.9 ± 4.1(1)	91.1 ± 2.6(1)	94.0 ± 2.3(0)	**94.8 ± 2.2**
16	91.5 ± 2.3(1)	86.1 ± 3.8(1)	83.5 ± 3.7(1)	83.9 ± 3.6(1)	86.6 ± 2.6(1)	84.2 ± 3.7(1)	90.4 ± 3.4(1)	**94.0 ± 2.0**(0)	93.9 ± 2.1
18	89.7 ± 2.4(1)	82.7 ± 3.5(1)	82.7 ± 2.4(1)	84.3 ± 3.8(1)	87.0 ± 3.2(1)	84.2 ± 3.7(1)	89.0 ± 3.0(1)	92.1 ± 1.4(1)	**93.2 ± 1.4**
20	89.2	85.3	85.8	80.8	85.6	85.9	88.9	90.7	**92.8**
Avg.	92.8	89.2	87.7	88.0	90.7	88.6	93.1	95.2	**96.4**
					Purity				
2	99.8 ± 0.3(0)	99.8 ± 0.3(0)	99.9 ± 0.3(0)	99.7 ± 0.6(1)	99.9 ± 0.3(0)	99.9 ± 0.3(0)	99.8 ± 0.3(0)	99.8 ± 0.3(0)	**99.9 ± 0.2**
4	99.2 ± 0.9(1)	98.9 ± 1.7(1)	95.5 ± 7.0(1)	97.1 ± 5.8(0)	97.9 ± 4.1(0)	97.9 ± 3.1(1)	99.3 ± 0.9(1)	99.6 ± 0.5(1)	**99.8 ± 0.3**
6	96.0 ± 5.2(1)	95.6 ± 5.5(1)	91.3 ± 5.9(1)	94.9 ± 5.4(1)	96.4 ± 5.2(1)	92.6 ± 6.3(1)	98.2 ± 2.2(1)	98.8 ± 2.1(0)	**99.4 ± 0.4**
8	94.8 ± 4.7(1)	90.0 ± 6.2(1)	89.0 ± 7.2(1)	88.7 ± 3.7(1)	93.3 ± 5.2(1)	91.8 ± 4.1(1)	97.3 ± 4.2(1)	98.4 ± 2.5(1)	**99.3 ± 1.3**
10	93.6 ± 4.4(1)	89.5 ± 5.8(1)	89.6 ± 5.4(1)	86.7 ± 5.1(1)	92.5 ± 4.7(1)	89.6 ± 4.5(1)	95.6 ± 3.0(1)	97.0 ± 3.2(1)	**99.0 ± 0.6**
12	90.6 ± 3.4(1)	86.3 ± 3.5(1)	82.7 ± 5.0(1)	84.8 ± 4.4(1)	90.1 ± 3.5(1)	86.3 ± 4.7(1)	94.2 ± 2.8(1)	95.4 ± 2.2(1)	**97.3 ± 2.1**
14	90.1 ± 3.6(1)	87.3 ± 3.7(1)	83.8 ± 3.8(1)	83.6 ± 3.6(1)	88.1 ± 4.0(1)	84.3 ± 3.5(1)	93.2 ± 2.5(1)	94.7 ± 2.5(0)	**95.6 ± 2.5**
16	91.2 ± 2.4(1)	85.2 ± 4.3(1)	81.8 ± 4.0(1)	81.9 ± 3.3(1)	86.7 ± 2.9(1)	83.5 ± 3.3(1)	92.4 ± 2.7(1)	94.4 ± 2.0(0)	**94.5 ± 2.3**
18	89.1 ± 2.8(1)	80.6 ± 4.0(1)	80.4 ± 2.4(1)	82.5 ± 4.5(1)	86.4 ± 3.5(1)	83.5 ± 3.3(1)	91.5 ± 2.2(1)	92.6 ± 1.4(0)	**93.3 ± 1.7**
20	87.6	85.0	83.1	77.7	84.5	82.2	91.5	91.7	**92.0**
Avg.	93.2	89.8	87.7	87.7	91.6	89.2	95.3	96.3	**97.0**

numbers. For each given cluster number c, 20 test runs were conducted on different randomly selected clusters (except the case when the whole data set is used). The mean and standard error of the performance are reported in the tables where the best results are highlighted in boldface.

From these five tables, we can observe that the proposed G-JNFC model obtains better performance than the other methods in most cases, reflecting its effectiveness in data clustering. In G-JNFC, the two regularization terms bring more desirable properties into the variables, *i.e.*, the basis matrix and the coefficient matrix; specifically, the second term in (6) builds bidirectional connectionship between the basis matrix and the coefficient matrix and the third term in (6) enforces the learned coefficient to respect the local manifold of data. By pairwisely comparing the results obtained by JNFC and NMF, we can conclude that the local coordinate coding is beneficial to boost the clustering performance. Similarly, since the results of G-JNFC are better than those of JNFC, meaning that exploring the structure information of data can is beneficial to efficient coefficient learning. It worths mentioning that though G-JNFC obtained weaker performance in some cases, it provides us a direct way to achieve the final clustering results and therefore there is no necessary to perform any postprocessing step. What is more, we may find the data points in boundary area based on the fuzzy clustering indicator.

To illustrate the statistical significance between our G-JNFC method and other algorithms, we performed the paired students t-test on their clustering results. Here the hypothesis is "the clustering performance (accuracy, normalized mutual information, purity) obtained by G-JNFC is greater than that obtained by the other (given) method". Each test was run on two sequences corresponding to clustering results of 20 splits by our method and the given method. In Tables 2–6, the statistical results were reported in which the binary value in brackets. For each entity in brackets, "1" means that the hypothesis is correct (true) with probability 0.95, and "0" means that the hypothesis is wrong (false) with probability 0.95. For example, on dig1-10 data set when the number of clusters is 9 in Table 4, the accuracy of GNLCF over 20 runs is 80.3 ± 4.9 and the accuracy of G-JNFC over 20 runs is 83.1 ± 3.8. The appended "(1)" means the hypothesis of G-JNFC is superior to GNLCF is correct based on the statistical test. In summary, from Tables 2–6, we see the conclusion that our G-JNFC achieves better performance is correct in most of cases on all the data sets.

As shown in (6), there are two regularization parameters in the G-JNFC objective function. To show how they affect the performance of G-JNFC in clustering, we show the variation of G-JNFC performance in terms of the three metrics on Caltech101 and TDT2 data sets by grid searching both parameters from $\{10^{-5}, 10^{-4}, \cdots, 10^{5}\}$ in Fig. 3. Obviously, there are wide ranges for the candidate values selection to parameter combination (λ, γ) to let G-JNFC achieve promising performance. Roughly, G-JNFC tends to select a slightly large λ and a slightly small γ, which further indicates the importance of the second term in (6). Similar landscapes can be found on the remaining two data sets. From this figure and the analysis, we can generally conclude that G-JNFC have stable performance with respect to parameters λ and γ.

Besides the theoretical analysis on the convergence of G-JNFC in Section 3.3, we empirically show the variation of its objective values in terms of iterations on the five used data sets in Fig. 4. From this figure, we observe that our model has desirable convergence speed, usually within 10 iterations.

Table 4
Clustering performance (%) of compared algorithms on dig1-10 data set.

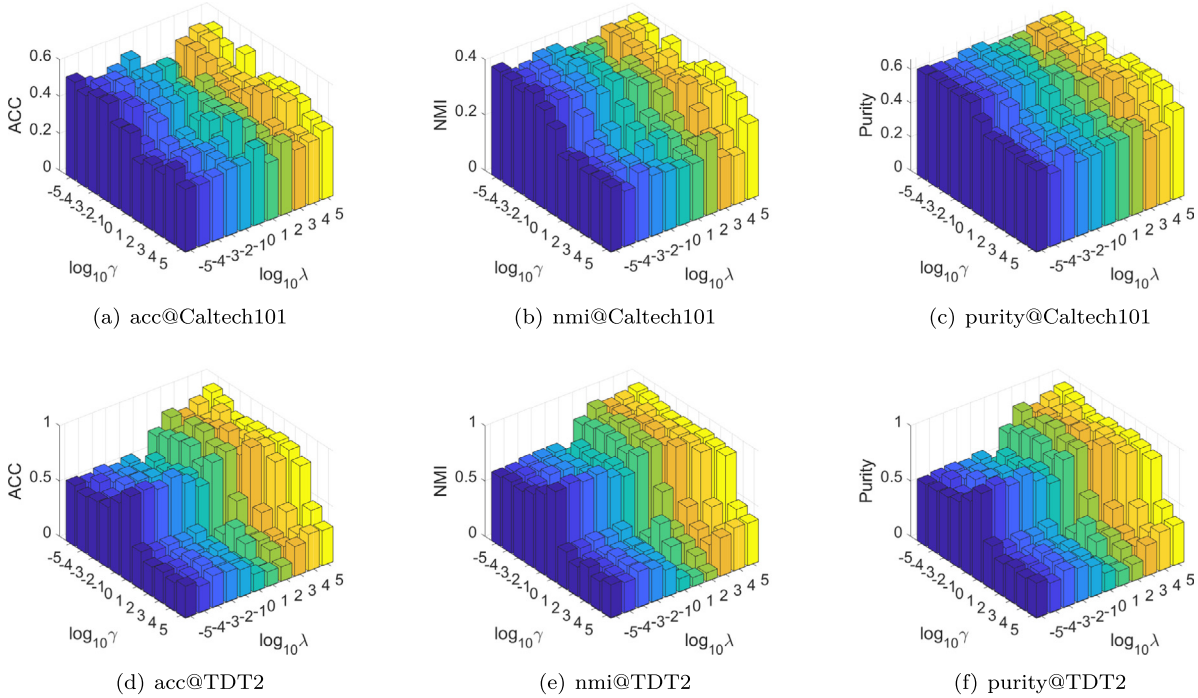
c	*K*means	NMF	CF	NCut	NLCF	GNLCF	FCM	JNFC	G-JNFC
					Accuracy				
2	99.1 ± 1.0(1)	98.2 ± 2.5(1)	98.3 ± 2.5(1)	95.6 ± 13.3(0)	98.9 ± 1.3(1)	99.3 ± 1.0(0)	98.9 ± 1.1(1)	99.2 ± 0.9(1)	**99.7 ± 0.5**
3	94.3 ± 3.9(1)	92.6 ± 4.2(1)	85.5 ± 15.2(1)	89.9 ± 14.1(0)	94.3 ± 3.6(1)	95.6 ± 3.0(0)	94.5 ± 3.5(1)	94.7 ± 3.7(1)	**96.0 ± 2.8**
4	90.6 ± 8.2(1)	86.5 ± 9.6(1)	82.0 ± 12.4(1)	89.6 ± 11.7(1)	90.4 ± 7.9(1)	92.9 ± 4.9(1)	91.2 ± 8.0(1)	92.4 ± 6.5(1)	**94.6 ± 3.7**
5	87.9 ± 9.5(1)	81.2 ± 10.3(1)	83.2 ± 9.4(1)	80.8 ± 10.7(1)	87.8 ± 8.8(1)	90.6 ± 6.5(1)	88.1 ± 9.4(1)	90.0 ± 6.4(1)	**92.6 ± 4.5**
6	85.7 ± 9.4(1)	80.2 ± 9.3(1)	75.1 ± 9.3(1)	83.3 ± 7.2(1)	86.6 ± 6.6(1)	86.8 ± 8.5(1)	87.3 ± 9.0(1)	88.9 ± 6.6(1)	**91.3 ± 5.8**
7	79.7 ± 6.6(1)	75.5 ± 6.8(1)	73.3 ± 5.6(1)	80.5 ± 9.3(1)	84.12 ± 4.2(1)	82.6 ± 4.7(1)	84.9 ± 6.0(1)	83.3 ± 5.7(1)	**88.9 ± 2.9**
8	80.6 ± 8.1(1)	74.8 ± 5.9(1)	71.7 ± 7.5(1)	79.3 ± 8.3(1)	80.13 ± 5.7(1)	81.6 ± 6.6(1)	83.6 ± 6.7(1)	81.3 ± 7.8(1)	**87.0 ± 4.7**
9	77.8 ± 4.1(1)	72.7 ± 4.5(1)	72.1 ± 5.7(1)	76.3 ± 5.3(1)	79.22 ± 5.0(1)	80.3 ± 4.9(1)	81.1 ± 4.6(1)	79.8 ± 4.6(1)	**83.1 ± 3.8**
10	70.0	76.9	74.4	62.6	76.9	74.3	79.4	83.8	**84.6**
Avg.	85.1	82.1	79.5	82.0	86.5	87.1	87.7	88.1	**90.9**
					Normalized Mutual Information				
2	93.5 ± 6.6(1)	89.6 ± 12.0(1)	90.1 ± 12.3(1)	90.2 ± 28.5(0)	92.8 ± 7.5(1)	95.4 ± 5.9(1)	92.7 ± 6.9(1)	94.0 ± 5.9(1)	**97.6 ± 3.5**
3	81.1 ± 10.8(1)	76.2 ± 12.0(1)	70.9 ± 18.4(1)	83.9 ± 18.7(0)	80.7 ± 10.4(1)	84.8 ± 8.8(0)	81.3 ± 10.2(1)	82.1 ± 10.4(1)	**85.8 ± 8.4**
4	79.5 ± 11.6(1)	71.3 ± 13.7(1)	68.8 ± 14.2(1)	**87.6 ± 11.9**(0)	77.7 ± 13.0(1)	81.0 ± 11.8(1)	79.9 ± 11.6(1)	80.9 ± 11.1(1)	85.1 ± 8.7
5	78.1 ± 11.0(1)	70.3 ± 10.6(1)	70.9 ± 10.1(1)	81.5 ± 9.4(0)	77.0 ± 10.6(1)	80.1 ± 11.2(0)	78.3 ± 11.2(1)	78.3 ± 11.0(1)	**83.0 ± 9.1**
6	77.6 ± 10.0(1)	70.0 ± 9.4(1)	67.7 ± 8.5(1)	**84.1 ± 6.4**(0)	76.6 ± 8.6(1)	77.2 ± 10.9(1)	78.7 ± 10.1(1)	79.1 ± 8.5(1)	82.4 ± 8.7
7	74.2 ± 5.2(1)	67.7 ± 5.2(1)	66.5 ± 4.7(1)	**82.7 ± 5.9**(1)	73.3 ± 3.7(1)	72.8 ± 6.4(1)	76.4 ± 5.1(1)	74.4 ± 5.9(1)	79.5 ± 4.6
8	74.9 ± 6.2(1)	67.3 ± 4.7(1)	66.2 ± 5.3(1)	**82.5 ± 5.9**(1)	72.1 ± 5.3(1)	73.2 ± 6.3(1)	75.7 ± 6.3(1)	73.9 ± 6.3(1)	78.0 ± 6.0
9	73.1 ± 2.6(1)	66.5 ± 3.5(1)	66.5 ± 3.5(1)	**81.7 ± 2.6**(1)	72.0 ± 4.4(1)	72.8 ± 5.2(1)	74.4 ± 3.7(1)	73.4 ± 3.5(1)	75.9 ± 3.8
10	70.9	70.1	67.5	74.2	70.3	70.3	73.5	75.0	**75.1**
Avg.	78.1	72.1	70.6	**83.2**	76.9	78.6	80.0	79.0	82.5
					Purity				
2	99.1 ± 1.0(1)	98.2 ± 2.5(1)	98.3 ± 2.5(1)	95.6 ± 13.3(0)	98.9 ± 1.3(1)	99.3 ± 1.0(0)	98.9 ± 1.1(1)	99.2 ± 0.9(1)	**99.7 ± 0.5**
3	94.3 ± 3.9(1)	92.6 ± 4.2(1)	86.7 ± 13.0(1)	90.1 ± 13.6(0)	94.3 ± 3.6(1)	95.6 ± 3.0(0)	94.5 ± 3.5(1)	94.7 ± 3.7(1)	**96.0 ± 2.8**
4	90.6 ± 8.2(1)	86.5 ± 9.6(1)	83.2 ± 11.07(1)	90.0 ± 11.1(1)	90.4 ± 7.9(1)	92.9 ± 5.0(1)	91.9 ± 6.6(1)	92.4 ± 6.5(1)	**94.6 ± 3.7**
5	88.2 ± 9.0(1)	82.1 ± 9.1(1)	84.0 ± 8.0(1)	83.5 ± 8.4(1)	88.0 ± 8.3(1)	90.6 ± 6.5(1)	89.5 ± 7.0(1)	90.0 ± 6.4(1)	**92.6 ± 4.5**
6	86.1 ± 8.9(1)	80.7 ± 8.9(1)	77.7 ± 7.6(1)	84.7 ± 6.1(1)	86.9 ± 6.3(1)	86.8 ± 8.5(1)	88.8 ± 6.9(1)	88.9 ± 6.5(1)	**91.3 ± 5.8**
7	81.1 ± 5.6(1)	77.3 ± 5.3(1)	75.9 ± 4.2(1)	83.7 ± 7.2(1)	84.4 ± 3.7(1)	86.6 ± 4.6(1)	86.4 ± 4.1(1)	83.4 ± 5.5(1)	**88.9 ± 2.9**
8	81.8 ± 6.8(1)	76.3 ± 4.9(1)	74.6 ± 5.5(1)	82.0 ± 6.5(1)	80.6 ± 5.4(1)	82.1 ± 6.0(1)	85.2 ± 5.1(1)	81.7 ± 7.3(1)	**87.0 ± 4.7**
9	78.7 ± 2.9(1)	74.0 ± 3.6(1)	74.2 ± 4.6(1)	80.3 ± 3.7(0)	79.6 ± 4.7(1)	80.7 ± 4.7(1)	83.0 ± 3.3(0)	80.2 ± 4.1(1)	**83.1 ± 3.8**
10	73.0	77.7	75.7	70.3	77.6	75.7	81.9	83.8	**84.6**
Avg.	85.9	82.8	81.1	84.5	86.7	87.8	89.9	88.3	**90.9**

Table 5
Clustering performance (%) of compared algorithms on Yale data set.

c	*K*means	NMF	CF	NCut	NLCF	GNLCF	FCM	JNFC	G-JNFC
					Accuracy				
2	80.2 ± 16.4(1)	66.1 ± 14.9(1)	67.5 ± 13.9(1)	82.1 ± 11.8(1)	87.3 ± 12.0(1)	90.5 ± 5.3(0)	83.2 ± 11.5(1)	87.5 ± 7.8(1)	**92.1 ± 7.5**
4	58.6 ± 9.2(1)	58.6 ± 7.9(1)	56.6 ± 9.1(1)	66.4 ± 10.6(1)	68.5 ± 9.6(1)	70.8 ± 9.2(1)	64.8 ± 11.0(1)	65.8 ± 9.2(1)	**76.6 ± 6.8**
6	46.5 ± 7.0(1)	48.6 ± 7.9(1)	46.3 ± 6.2(1)	52.0 ± 8.4(1)	55.3 ± 6.8(1)	56.7 ± 7.0(1)	50.4 ± 6.3(1)	51.7 ± 5.4(1)	**61.7 ± 6.2**
8	45.2 ± 5.0(1)	47.1 ± 5.0(1)	43.6 ± 7.0(1)	49.4 ± 5.0(1)	53.4 ± 4.1(1)	53.1 ± 3.9(1)	50.7 ± 4.5(1)	49.1 ± 4.2(1)	**57.4 ± 5.8**
10	45.6 ± 5.8(1)	44.1 ± 4.4(1)	43.64 ± 4.9(1)	49.2 ± 4.5(1)	49.4 ± 4.4(1)	49.3 ± 5.1(1)	48.5 ± 4.6(1)	44.1 ± 4.5(1)	**53.7 ± 3.9**
12	42.8 ± 3.6(1)	40.2 ± 4.1(1)	36.70 ± 4.0(1)	44.4 ± 3.7(1)	46.9 ± 2.8(0)	45.0 ± 3.2(1)	44.5 ± 3.9(1)	41.1 ± 3.1(1)	**47.6 ± 3.2**
15	38.8	37.6	40.0	39.4	**45.5**	41.8	47.9	39.4	42.4
Avg.	51.1	48.9	47.8	54.7	58.0	58.2	55.7	54.1	**61.6**
					Normalized Mutual Information				
2	42.1 ± 30.7(1)	16.5 ± 24.7(1)	17.3 ± 24.4(1)	41.5 ± 27.5(1)	55.2 ± 28.8(1)	59.8 ± 18.1(0)	42.5 ± 26.7(1)	52.5 ± 24.4(1)	**67.7 ± 26.5**
4	40.4 ± 11.9(1)	40.8 ± 11.6(1)	36.5 ± 13.4(1)	46.3 ± 10.6(1)	49.1 ± 13.4(1)	50.3 ± 13.1(1)	44.9 ± 15.9(1)	45.4 ± 14.0(1)	**56.0 ± 10.1**
6	35.8 ± 8.4(1)	37.3 ± 8.7(1)	34.3 ± 7.3(1)	41.3 ± 9.8(1)	43.4 ± 7.8(1)	44.3 ± 8.9(1)	38.0 ± 6.9(1)	38.6 ± 8.8(1)	**49.1 ± 7.8**
8	40.9 ± 6.0(1)	41.8 ± 5.1(1)	37.06 ± 6.8(1)	44.2 ± 5.2(1)	47.9 ± 4.8(0)	45.5 ± 5.4(1)	45.1 ± 3.7(1)	43.2 ± 5.1(1)	**49.7 ± 6.4**
10	44.7 ± 5.9(1)	43.5 ± 3.3(1)	41.95 ± 4.9(1)	48.0 ± 4.7(1)	46.9 ± 4.2(1)	47.3 ± 5.5(1)	47.5 ± 4.6(1)	41.5 ± 5.4(1)	**50.1 ± 4.7**
12	45.0 ± 4.3(1)	42.9 ± 3.1(1)	39.58 ± 3.3(1)	46.9 ± 3.4(0)	48.1 ± 3.2(0)	44.9 ± 3.7(1)	46.3 ± 3.5(1)	41.4 ± 4.1(1)	**48.1 ± 4.4**
15	44.9	44.4	42.6	48.6	49.7	43.9	**51.3**	43.0	48.0
Avg.	42.0	38.2	35.6	45.2	48.6	45.1	45.1	43.7	**52.7**
					Purity				
2	80.2 ± 16.4(1)	66.1 ± 14.9(1)	67.5 ± 13.9(1)	82.1 ± 11.8(1)	87.3 ± 12.0(1)	90.5 ± 5.3(0)	83.2 ± 11.5(1)	87.5 ± 7.8(1)	**92.1 ± 7.5**
4	59.8 ± 9.2(1)	59.7 ± 8.2(1)	57.3 ± 9.3(1)	67.1 ± 9.7(1)	68.5 ± 9.6(1)	71.0 ± 8.9(1)	72.5 ± 9.2(1)	66.3 ± 9.1(1)	**76.7 ± 6.6**
6	48.1 ± 6.6(1)	49.6 ± 7.4(1)	47.2 ± 5.9(1)	53.1 ± 8.1(1)	55.7 ± 6.4(1)	57.4 ± 7.0(1)	60.6 ± 5.7(1)	52.2 ± 5.3(1)	**62.5 ± 5.9**
8	47.0 ± 4.9(1)	48.4 ± 4.8(1)	44.2 ± 6.5(1)	50.7 ± 5.5(1)	53.8 ± 4.0(0)	53.9 ± 3.8(1)	55.1 ± 4.5(1)	49.8 ± 4.0(1)	**58.0 ± 5.9**
10	46.8 ± 5.4(1)	45.7 ± 4.0(1)	44.6 ± 4.7(1)	49.9 ± 4.5(1)	49.8 ± 4.6(1)	50.1 ± 4.9(1)	**54.9 ± 4.2**(0)	45.0 ± 4.42(1)	54.1 ± 4.1
12	44.1 ± 3.7(1)	41.9 ± 3.4(1)	38.5 ± 3.5(1)	45.5 ± 3.6(1)	47.8 ± 3.0(0)	45.3 ± 3.3(1)	47.3 ± 3.8(0)	42.1 ± 3.1(1)	**48.3 ± 3.3**
15	41.2	38.8	40.0	40.0	46.1	41.8	**50.3**	41.2	43.0
Avg.	52.5	50.0	48.5	55.5	58.4	58.6	60.6	54.9	**62.1**

Table 6
Clustering performance (%) of compared algorithms on Isolet data set.

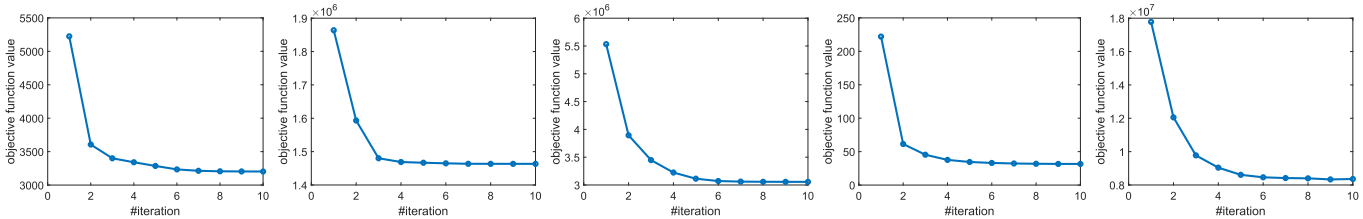
c	*K*means	NMF	CF	NCut	NLCF	GNLCF	FCM	JNFC	G-JNFC
					Accuracy				
4	93.2 ± 11.2(1)	87.6 ± 11.7(1)	83.0 ± 12.0(1)	92.4 ± 11.7(1)	95.8 ± 5.4(1)	96.5 ± 5.5(1)	96.5 ± 5.3(1)	95.9 ± 7.2(0)	**98.0 ± 3.8**
8	77.9 ± 9.7(1)	72.7 ± 9.6(1)	66.2 ± 7.7(1)	72.9 ± 11.4(1)	77.5 ± 7.6(1)	79.8 ± 9.4(1)	82.0 ± 9.0(1)	77.4 ± 8.1(1)	**83.3 ± 8.1**
12	75.8 ± 7.3(1)	73.4 ± 9.1(1)	63.7 ± 7.9(1)	70.2 ± 7.0(1)	80.4 ± 6.6(0)	78.6 ± 7.5(1)	75.6 ± 8.1(1)	75.2 ± 8.3(1)	**80.8 ± 4.5**
16	69.7 ± 5.8(1)	69.1 ± 5.6(1)	54.3 ± 4.9(1)	64.4 ± 5.9(1)	**74.3 ± 4.2**(1)	72.1 ± 5.8(1)	70.6 ± 5.9(1)	69.0 ± 6.9(1)	**74.3 ± 4.4**
20	65.5 ± 4.6(1)	64.4 ± 5.0(1)	47.6 ± 4.7(1)	57.6 ± 4.7(1)	69.2 ± 5.0(1)	67.8 ± 4.8(1)	67.8 ± 6.9(1)	63.9 ± 4.3(1)	**69.4 ± 4.1**
24	63.1 ± 3.5(1)	61.3 ± 3.1(1)	43.9 ± 2.9(1)	55.1 ± 4.4(1)	65.7 ± 2.8(1)	63.7 ± 3.0(1)	64.5 ± 2.1(0)	59.7 ± 3.7(1)	**65.8 ± 2.4**
26	60.6	55.6	39.0	50.5	58.5	61.4	63.9	64.7	**67.3**
Avg.	72.2	69.2	56.8	66.1	74.5	74.2	74.4	72.3	**77.0**
					Normalized Mutual Information				
4	90.6 ± 10.0(1)	83.0 ± 11.3(1)	77.8 ± 12.3(1)	90.2 ± 11.2(1)	91.4 ± 8.8(1)	92.0 ± 9.7(1)	92.6 ± 8.1(1)	92.4 ± 9.3(1)	**95.3 ± 6.0**
8	79.0 ± 8.3(1)	74.1 ± 8.7(1)	68.1 ± 8.1(1)	78.3 ± 8.9(1)	78.2 ± 6.0(1)	78.1 ± 8.1(1)	79.0 ± 8.1(1)	79.5 ± 8.2(1)	**81.8 ± 7.2**
12	81.6 ± 4.8(1)	79.0 ± 6.4(1)	68.5 ± 5.0(1)	79.5 ± 5.2(1)	83.1 ± 4.4(0)	81.6 ± 5.6(1)	82.3 ± 6.1(1)	80.5 ± 6.5(1)	**84.0 ± 4.2**
16	79.5 ± 3.8(0)	77.5 ± 3.8(1)	64.2 ± 3.7(1)	77.4 ± 3.8(1)	80.8 ± 2.8(1)	78.7 ± 3.7(1)	79.9 ± 3.4(0)	78.2 ± 4.9(1)	**80.6 ± 3.5**
20	77.0 ± 2.7(1)	74.8 ± 2.9(1)	60.2 ± 3.7(1)	73.6 ± 3.1(1)	77.8 ± 3.1(1)	76.6 ± 2.8(1)	77.6 ± 3.1(0)	75.5 ± 3.3(1)	**78.3 ± 2.6**
24	76.2 ± 1.5(1)	74.5 ± 1.0(1)	59.3 ± 2.2(1)	73.2 ± 2.8(1)	76.7 ± 1.9(1)	75.0 ± 1.9(1)	75.4 ± 1.5(1)	74.5 ± 2.1(1)	**77.2 ± 1.3**
26	75.2	70.9	54.6	74.4	75.9	73.1	75.1	75.5	**77.5**
Avg.	79.9	76.2	64.7	78.1	80.5	79.3	80.2	79.5	**82.1**
					Purity				
4	94.3 ± 8.5(1)	88.7 ± 9.8(1)	84.4 ± 10.2(1)	93.4 ± 9.3(1)	95.8 ± 5.4(1)	96.5 ± 5.5(1)	96.5 ± 5.3(1)	96.0 ± 6.9(0)	**98.0 ± 3.8**
8	79.9 ± 8.8(1)	75.1 ± 8.7(1)	69.3 ± 7.9(1)	76.9 ± 9.8(1)	79.4 ± 6.7(0)	80.4 ± 8.7(1)	82.4 ± 8.6(1)	79.4 ± 7.7(1)	**83.6 ± 7.7**
12	79.3 ± 6.0(1)	76.5 ± 8.2(1)	66.1 ± 7.1(1)	74.8 ± 5.9(1)	82.2 ± 5.9(0)	80.5 ± 6.7(0)	**83.9 ± 7.1**(0)	76.9 ± 8.5(1)	82.8 ± 4.2
16	73.7 ± 5.2(1)	72.6 ± 5.5(1)	56.7 ± 4.5(1)	69.3 ± 4.8(1)	**77.4 ± 3.7**(1)	74.8 ± 5.0(1)	76.4 ± 5.3(0)	71.7 ± 7.2(1)	76.4 ± 4.7
20	69.8 ± 4.3(1)	67.7 ± 4.4(1)	50.3 ± 4.1(1)	62.8 ± 4.4(1)	72.1 ± 4.3(0)	70.8 ± 4.2(0)	**72.8 ± 3.2**(0)	66.7 ± 4.8(1)	71.8 ± 4.0
24	67.2 ± 3.0(1)	65.6 ± 2.0(1)	46.5 ± 2.8(1)	60.1 ± 3.9(1)	**69.2 ± 2.7**(1)	67.0 ± 3.0(1)	68.5 ± 2.2(0)	62.9 ± 3.7(1)	68.4 ± 2.4
26	64.9	59.9	40.7	57.9	65.4	64.2	68.7	68.7	**68.8**
Avg.	75.6	72.3	59.1	70.8	77.3	76.3	78.4	74.6	**78.5**



(a) acc@Caltech101 (b) nmi@Caltech101 (c) purity@Caltech101

(d) acc@TDT2 (e) nmi@TDT2 (f) purity@TDT2

Fig. 3. The performance of G-JNFC in terms of parameters λ and γ on Caltech101 and TDT2 data sets.



(a) Caltech101 (b) TDT2 (c) dig1-10 (d) Yale (e) Isolet

Fig. 4. Convergence property of G-JNFC on the five used data sets.

5. Conclusion

In this paper, we proposed a joint non-negative and fuzzy coding with graph regularization model, termed G-JNFC in which the learned coefficient matrix has two roles. One is that it can be viewed as the data representation and the other is that it is the fuzzy membership degree matrix. Based on the learned coefficient matrix, we can not only determine the membership of each data point to a specified cluster, but also can obtain the membership degree of each data point to different clusters if it is near the clustering boundary. Compared with existing matrix factorization based clustering methods, G-JNFC effectively avoids the limitations in the two-stage clustering methods. Finally, we perform extensive experiments to demonstrate the effectiveness of G-JNFC.

Funding

This work was partially supported by Natural Science Foundation of China (61971173, 61602140, U1909202, 61972121), China Postdoctoral Science Foundation (2017M620470), Provincial Key Laboratory for Computer Information Processing Technology, Soochow University (KJS1841), Planted Talent Plan of Zhejiang Province (2019R407030) and Open Fund of Engineering Research Center of Cognitive Healthcare of Zhejiang Province, Sir Run Run Shaw Hospital (2018KFJJ05).

References

[1] Wang Y-X, Zhang Y-J. Nonnegative matrix factorization: a comprehensive review. IEEE Trans Knowl Data Eng 2012;25(6):1336–53.
[2] He Y-C, Lu H-T, Huang L, Shi X-H. Non-negative matrix factorization with pairwise constraints and graph laplacian. Neural Process Lett 2015;42(1):167–85.
[3] Peng Y, Long Y, Qin F, Kong W, Cichocki A. Flexible non-negative matrix factorization with adaptively learned graph regularization. In: IEEE International Conference on Acoustics, Speech and Signal Processing. p. 3107–11.
[4] Zhang Z, Zhang Y, Li S, Liu G, Zeng D, Yan S, Wang M. Flexible auto-weighted local-coordinate concept factorization: A robust framework for unsupervised clustering. IEEE Trans Knowled Data Enghttps://doi.org/10.1109/TKDE.2019.2940576..
[5] Li Z, Tang J, He X. Robust structured nonnegative matrix factorization for image representation. IEEE Trans Neural Networks Learn Syst 2018;29(5):1947–60.
[6] Zhang Z, Zhang Y, Liu G, Tang J, Yan S, Wang M. Joint label prediction based semi-supervised adaptive concept factorization for robust data representation. IEEE Trans Knowl Data Eng 2020;32(5):952–70.
[7] Long X, Lu H, Peng Y, Li W. Graph regularized discriminative non-negative matrix factorization for face recognition. Multimedia Tools Appl 2014;72(3):2679–99.
[8] Wu C, Song Y, Zhang Y. Multi-view gait recognition using NMF and 2DLDA. Multimedia Tools Appl 2019;78(24):35789–811.
[9] Fodeh SJ, Tiwari A. Exploiting medline for gene molecular function prediction via nmf based multi-label classification. J Biomed Inform 2018;86:160–6.
[10] Wang M, Zhang B, Pan X, Yang S. Group low-rank nonnegative matrix factorization with semantic regularizer for hyperspectral unmixing. IEEE J Selected Top Appl Earth Observ Remote Sens 2018;11(4):1022–9.
[11] Elhamifar E, Vidal R. Sparse subspace clustering. IEEE Conference on Computer Vision and Pattern Recognition 2009:2790–7.
[12] Elhamifar E, Vidal R. Sparse subspace clustering: algorithm, theory, and applications. IEEE Trans Pattern Anal Mach Intell 2013;35(11):2765–81.
[13] Peng Y, Lu B-L. Discriminative extreme learning machine with supervised sparsity preserving for image classification. Neurocomputing 2017;261:242–52.
[14] Liu G, Lin Z, Yu Y. Robust subspace segmentation by low-rank representation. In: International Conference on International Conference on Machine Learning. p. 663–70.
[15] Liu G, Lin Z, Yan S, Sun J, Yu Y, Ma Y. Robust recovery of subspace structures by low-rank representation. IEEE Trans Pattern Anal Mach Intell 2013;35(1):171–84.
[16] Peng Y, Lu B-L, Wang S. Enhanced low-rank representation via sparse manifold adaption for semi-supervised learning. Neural Networks 2015;65:1–17.
[17] Simhachalam B, Ganesan G. Performance comparison of fuzzy and non-fuzzy classification methods. Egypt Inform J 2016;17(2):183–8.
[18] Alruwaili M, Siddiqi MH, Javed MA. A robust clustering algorithm using spatial fuzzy C-means for brain MR images. Egypt Inf J 2020;21(1):51–66.
[19] Yu K, Zhang T, Gong Y. Nonlinear learning using local coordinate coding. Advances in neural information processing systems 2009:2223–31.
[20] Chen Y, Zhang J, Cai D, Liu W, He X. Nonnegative local coordinate factorization for image representation. IEEE Trans Image Process 2013;22(3):969–79.
[21] Liu H, Yang Z, Yang J, Wu Z, Li X. Local coordinate concept factorization for image representation. IEEE Trans Neural Networks Learn Syst 2013;25(6):1071–82.
[22] Zhang R, Nie F, Guo M, Wei X, Li X. Joint learning of fuzzy k-means and nonnegative spectral clustering with side information. IEEE Trans Image Process 2019;28(5):2152–62.
[23] Cai D, He X, Han J, Huang TS. Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 2010;33(8):1548–60.
[24] Kyrillidis A, Becker S, Cevher V, Koch C. Sparse projections onto the simplex. In: International Conference on Machine Learning. p. 235–43.
[25] Nie F, Yang S, Zhang R, Li X. A general framework for auto-weighted feature selection via global redundancy minimization. IEEE Trans Image Process 2018;28(5):2428–38.
[26] Chen X, Yuan G, Nie F, Zhong M. Semi-supervised feature selection via sparse rescaled linear square regression. IEEE Trans Knowl Data Eng 2020;32(1):165–76.
[27] Sherman AH. On newton-iterative methods for the solution of systems of nonlinear equations. SIAM J Numer Anal 1978;15(4):755–71.
[28] Delbos F, Gilbert JC. Global linear convergence of an augmented lagrangian algorithm to solve convex quadratic optimization problems. J Convex Anal 2005;12(1):45–69.
[29] Huang J, Nie F, Huang H. A new simplex sparse learning model to measure data similarity for clustering. In: International Joint Conference on Artificial Intelligence. p. 3569–75.
[30] Xu W, Liu X, Gong Y. Document clustering based on non-negative matrix factorization. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 267–73.
[31] Xu W, Gong Y. Document clustering by concept factorization. In: International ACM SIGIR Conference on Research and Development in Information Retrieval. p. 202–9.