Constrained Dual Graph Regularized Orthogonal Nonnegative Matrix Tri-Factorization for Co-Clustering

Shaodi Ge, Hongjun Li, and Liuhong Luo
College of Science, Beijing Forestry University, Beijing 100083, China
Correspondence should be addressed to Liuhong Luo; llh7667@bjfu.edu.cn
Received 21 August 2019; Revised 4 December 2019; Accepted 9 December 2019; Published 26 December 2019

Academic Editor: Michele Betti
Copyright © 2019 Shaodi Ge et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Coclustering approaches for grouping data points and features have recently been receiving extensive attention. In this paper, we propose a constrained dual graph regularized orthogonal nonnegative matrix trifactorization (CDONMTF) algorithm to solve the coclustering problems. The new method improves the clustering performance obviously by employing hard constraints to retain the priori label information of samples, establishing two nearest neighbor graphs to encode the geometric structure of data manifold and feature manifold, and combining with biorthogonal constraints as well. In addition, we have also derived the iterative optimization scheme of CDONMTF and proved its convergence. Clustering experiments on 5 UCI machine-learning data sets and 7 image benchmark data sets show that the achievement of the proposed algorithm is superior to that of some existing clustering algorithms.

1. Introduction

Clustering analysis divides data sample points into clusters according to their similarity, and it has been commonly applied in data mining [1–4], machine learning [5], and image processing [6]. And coclustering can cluster rows and columns of the data matrix simultaneously, thereby capturing the binary relationship between data and features and obtaining better clustering results [7, 8].

The effectiveness of nonnegative matrix factorization (NMF) [9] has been proved to be effective in solving clustering and coclustering problems [10–13]. A nonnegative data matrix is decomposed into two or three nonnegative factors, and their product approximately linearizes the original matrix. Since the size of each factor matrix is much smaller than the original matrix, NMF provides a parts-based new representation to the original data [9, 14, 15]. Due to the nonnegative constraints of factor matrices, NMF with the additive combination can approximately match the procedure of processing raw data by human cognition and usually produces sparse data representation to effectively mine the potential structure of data [16, 17].

According to the needs of practical clustering tasks, several improved methods based on NMF are proposed. Deng et al. [18] put forward an unsupervised learning algorithm called graph-regularized NMF (GNMF) that creates a nearest neighbor graph to retain the neighborhood information of high-dimensional data in the low-dimensional representation space. It is worthy to be mentioned that graph Laplacian regularizer has been included in various NMF frameworks [19–21] after its first application in GNMF. Peng et al. [22] proposed NMF with integrated graph and feature learning (NMF2L) by introducing an error matrix to distinguish features and constructing a data graph to integrate feature learning and manifold learning. Shang et al. [23] presented an algorithm named as local discriminative based sparse subspace learning for feature selection, which introduces a local discriminant model into the feature selection framework of subspace learning. In addition, many researchers have found that semisupervised algorithms can enhance learning accuracy even with only a small amount of label information [24–26]. On this basis, Liu et al. introduced a constrained NMF (CNMF) algorithm for the scenario of limited label data [27], in which existing label information could serve as an additional hard constraint. Meng et al. [28] proposed a semisupervised graph regularized deep NMF with biorthogonal constraints (SGDNMF). SGDNMF can study the hidden layer
representation according to the unknown and various attributes of the original high-dimensional data to improve the effect of dimensionality reduction and the clustering performance simultaneously.

However, the aforementioned approaches focus mainly on one-side clustering, i.e., grouping similar objects. Co-clustering methods involve the relationship between features and data points to group the data of columns (data) and rows (features) at the same time [29–33]. Del Buono and Pio [34] improved the average intercluster similarity by optimizing block matrix. Shang et al. [35] developed a graph dual regularization NMTF (DNMTF) algorithm that constructs two associated graphs under the objective function of NMF and achieves better clustering performance. Meng et al. [36] proposed a dual-graph regularized NMF with sparse and orthogonal constraints (SODNMF) and obtained encouraging clustering results. Wang and Huang [7] suggested the penalized NMF algorithm (PNMT), in which three penalty terms are introduced to ensure that the clustering indicator matrix is close to orthogonality. Salehani et al. [19] combined graph regular constraint with NMF and proposed a clustering method called DNMTF. The main highlights of the proposed method are as follows:

(i) The priori label information of original data samples, intrinsic geometric structures in both data manifold and feature manifold, and biorthogonal constraints are integrated into a unified framework

(ii) An improved optimization problem is constructed by adopting the priori label information, geometric structures of data and feature, and biorthogonal constraints as well

(iii) The multiplicative update rules are derived to solve the optimization problem, and it is theoretically proved that the objective function does not increase under the update rules

(iv) The performance of the proposed algorithm is verified by clustering experiments on some open data sets

The following sections of this paper are presented as follows. Section 2 briefly reviews the related works and explicitly describes the proposed CDONMTF algorithm, including the multiplicative update rules and convergence analysis. Section 3 includes the clustering experiments and the result analysis. Finally, the summary is made in Section 4.

2. CDONMTF

This section reviews several NMF-based algorithms and describes the proposed CDONMTF algorithms in detail. We then derive the multiplicative iterative rules for the corresponding optimization problem and analyze its convergence.

2.1. Related Works. Given a nonnegative data matrix \( X \in \mathbb{R}^{m \times n} \), in which \( x_{ij} \in \mathbb{R}^{m \times n}, i = 1, 2, \ldots, n \), are \( m \)-dimensional nonnegative column vectors, typically representing data points. \( y_i = X_i^T \in \mathbb{R}^{m \times 1}, i = 1, 2, \ldots, m \), are \( n \)-dimensional nonnegative row vectors, typically representing features. The co-clustering method can group the data space into \( p \) clusters, as well as the feature space into \( q \) clusters [7].

A NMF algorithm can find two nonnegative matrices \( U \in \mathbb{R}^{m \times r} \) and \( V \in \mathbb{R}^{n \times r} \), whose product \( UV^T \) can be approximate to \( X \). It can be further understood that individual original data point \( x_i \) can be explicated as a linear combination of column vectors in \( U \), which can be thought of as a base matrix, and matrix \( V^T \) is an encoding matrix.

The error between the matrix \( X \) and the product \( UV^T \) is defined as a metric function \( D_{\text{NMF}}(X \| UV^T) \), which should be minimized and usually called an objective function. A commonly used objective function is based on the square of the Frobenius norm [9]:

\[
D_{\text{NMF}}(X \| UV^T) = \| X - UV^T \|_F^2 \quad \text{s.t. } U \geq 0, V \geq 0. \tag{1}
\]

Shang et al. [35] considered the geometric structure of both data manifold and feature manifold by establishing two associated graphs embedded in the objective function of NMF and proposed a clustering method called DNMTF. The objective function is as follows:

\[
D_{\text{DNMTF}}(X \| USV^T) = \| X - USV^T \|_F^2 + \alpha \text{Tr}(V^T L_u V) + \beta \text{Tr}(U^T L_v U) \quad \text{s.t. } U \geq 0, S \geq 0, V \geq 0, \tag{2}
\]

where \( S \) is an auxiliary matrix, \( \text{Tr}(\cdot) \) denotes the trace of the matrix, \( L_v \) and \( L_u \) are graph Laplacians of the data graph and the feature graph, respectively, and \( \alpha \) and \( \beta \) are regularized parameters.

It is usually infeasible to obtain labels for all samples in a data set, but the cost of obtaining a small portion of data labels is relatively low. Under this consideration, Liu et al. [27] extended NMF to a semisupervised learning algorithm called constrained NMF (CNMF). The objective function of CNMF is

\[
D_{\text{CNMF}}(X \| UV^T) = \| X - UZ^T A^T \|_F^2 \quad \text{s.t. } U \geq 0, Z \geq 0, \tag{3}
\]

where \( A \) represents a hard constraints matrix and \( Z \) is an auxiliary matrix.

2.2. Problem Formulation. DNMTF maintains the geometric structure of data manifold and feature manifold through two nearest neighbor graphs, and CNMF makes use of priori label information of original data effectively through hard constraints matrix. The clustering results of these two methods are much better than that of classical NMF methods. We try to integrate the idea of graph embedding...
and semisupervised learning and utilize biorthogonal constraints as an independent penalty to build the CDONMTF algorithm.

Firstly, we construct a data graph and a feature graph to simulate the geometric structures of data space and feature space [37]. Following the graph embedding formula, the inner compactness of data is characterized by the nearest neighbor graph. $S_k$ is used to measure the smoothness of $x_i$ and $x_j$ in the low-dimensional representation space:

$$
S_k = \frac{1}{2} \sum_{i,j=1}^{n} \left( f_k(x_i) - f_k(x_j) \right)^2 W^V_{ij}
$$

$$
= \sum_{i,j=1}^{n} f_k(x_i)^2 D^V_{ii} - \sum_{i,j=1}^{n} f_k(x_i) f_k(x_j) W^V_{ij}
$$

$$
= \sum_{i,j=1}^{n} v_i^T D^V_{ii} v_i - \sum_{i,j=1}^{n} v_i^T v_j W^V_{ij}
$$

$$
= \text{Tr}(V^T D^V V) - \text{Tr}(V^T W^V V)
$$

$$
= \text{Tr}(V^T L^V V),
$$

where $f_k(x_i) = v_i$ is the mapping function that maps the $x_i$ to its corresponding representation $v_i$ in the low-dimensional space. $W^V$ is the weight matrix to measure the pairwise similarity between the original data points, $D^V$ is a diagonal matrix defined by $D^V_{ii} = \sum_j W^V_{ij}$, and $L^V = D^V - W^V$ [16] is the Laplacian matrix of data graph. $S_k$ should be minimized to ensure the smoothness of data in the low-dimensional spatial relationship and make $v_i$ and $v_j$ as close as possible.

The construction of weight matrix $W^V$ is an important factor affecting the performance of algorithms. In recent years, many scholars have proposed different methods to assign values to $W^V$, such as robust adaptive neighbor [38], point product weighting [18], and kernel method [39, 40]. For simplicity, we use the 0-1 weighted scheme, but it is easy to be extended to other methods. Here, a k-nearest neighbor data graph with vertices corresponding to $\{x_1, x_2, \ldots, x_m\}$ is constructed and the following data weight matrix is defined:

$$
W^V_{ij} = \begin{cases} 
1, & \text{if } x_i \in N_k(x_j) \text{ or } x_j \in N_k(x_i), \\
0, & \text{otherwise},
\end{cases}
$$

where $N_k(x_j)$ represents the set of k-nearest neighbors of data point $x_j$.

Like the construction of the data graph, a k-nearest neighbor feature graph is also constructed by using the 0-1 weighted scheme and its vertices correspond to $\{y_1, y_2, \ldots, y_m\}$. Then, the feature weight matrix is defined:

$$
W^U_{ij} = \begin{cases} 
1, & \text{if } y_i \in N_k(y_j) \text{ or } y_j \in N_k(y_i), \\
0, & \text{otherwise},
\end{cases}
$$

where $D^U$ is a diagonal matrix and $D^U = \sum_j W^U_{ij}$, and $L^U = D^U - W^U$ is the Laplacian matrix of the feature graph. Hence, $\text{Tr}(U^T L^U U)$ is obtained as the regularizer term.

Assume that $n$ data points $\{x_i, i = 1, 2, \ldots, n\}$ belong to $c$ classes, in which the first $l$ samples are labeled and the remaining $n - l$ samples are unlabeled. We can construct an indicator matrix $C \in R^{c \times n}$:

$$
C_{ij} = \begin{cases} 
1, & \text{if } x_i \text{ is labeled with } j^{\text{th}} \text{ class}, \\
0, & \text{otherwise}.
\end{cases}
$$

Then, a label hard constraints matrix $A$ based on $C$ is defined:

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

where $A \in R^{(c \times n) \times (n-l)}$ and $I_{n-l}$ is an $(n-l) \times (n-l)$ identity matrix.

Combining the above two aspects with independent biorthogonal constraints, we can get the following objective function of CDONMTF:

$$
D_{CDONMTF}(X \parallel USV^	op) = \|X - USV^	op\|_F^2 + \alpha(\text{Tr}(V^T L^V V)) + \lambda \text{Tr}(S^T S),
$$

s.t. $U \geq 0, S \geq 0, V \geq 0, V = AZ, U^T U = I, V^T V = I$,

$$
(9)
$$

where $U \in R^{n \times p}$, $S \in R^{c \times q}$, and $V \in R^{q \times q}$, $\alpha$ and $\lambda$ are nonnegative regularized parameters, and $\text{Tr}(S^T S)$ prevents all elements in $U$ and $V$ being too small.

2.3. Optimization and Update Rules. In order to solve the optimization problem, we first convert equation (9) to the following form:

$$
D_{CDONMTF}(X \parallel USV^	op) = \|X - USZ^T A^T\|_F^2 + \lambda \text{Tr}(S^T S)
$$

$$
+ \alpha(\text{Tr}(Z^T A^T L^V V + \text{Tr}(U^T L^V U)))
$$

$$
+ \beta(\text{Tr}(A Z E Z^T A^T) + \text{Tr}(U E U^T)),
$$

s.t. $U \geq 0, S \geq 0, Z \geq 0$,

$$
(10)
$$

where $E = I - I(I)$ is a matrix of ones and $I$ is an identity matrix, $\text{Tr}(U E U^T) = \sum_{i,j} u_{ij} u_{ij}$ and $\text{Tr}(V E V^T) = \sum_{i,j} v_{ij} v_{ij} (V = AZ)$ are penalty items to ensure the orthogonality of $U$ and $V$, and $\beta$ is a nonnegative orthogonal constraint balance parameter.

The objective function equation (10) is not convex for $U$, $S$, and $Z$ together. Therefore, it is unrealistic to find a global minimum. So, we try to deduce the multiplicative update rules to look for a local minimum.

Let $\Psi = [\psi_{ij}], Y = [y_{ij}]$, and $\Phi = [\phi_{ij}]$ be the Lagrange multipliers of $U_{ij} \geq 0, S_{ij} \geq 0$, and $Z_{ij} \geq 0$, respectively. By using matrix properties $\text{Tr}(A B) = \text{Tr}(B A)$ and $\text{Tr}(A) = \text{Tr}(A^T)$, the Lagrangian function $\zeta$ is
\[ \zeta = D_{\text{CDONMTf}}(X\|USV^T) + \text{Tr}(\Psi U^T) + \text{Tr}(YS^T) + \text{Tr}(\Phi Z^T) \]
\[ = \text{Tr}(XX^T) - 2\text{Tr}(XAZS^TU^T) + \text{Tr}(USZ^TAXZ^TU^T) \]
\[ + a(\text{Tr}(Z^TA^TL_vA) + \text{Tr}(U^T L_vU)) + \lambda \text{Tr}(S^T S) \]
\[ + \beta(\text{Tr}(AZ^TA^T) + \text{Tr}(UEU^T)) + \text{Tr}(\Psi U^T) \]
\[ + \text{Tr}(YS^T) + \text{Tr}(\Phi Z^T). \]
\[ (11) \]

The partial derivatives of \( \zeta \) with respect to \( U, Z \) and \( S \) are
\[ \frac{\partial \zeta}{\partial U} = -2XAZS^T + 2USZ^T A^T AZS^T + 2aL_vU + 2\beta UE + \Psi, \]
\[ \frac{\partial \zeta}{\partial S} = -2U^TXAZ + 2U^TUSZ^T A^T AZ + 2\lambda S + Y, \]
\[ \frac{\partial \zeta}{\partial Z} = -2A^TX^TU + 2A^TAZ^TU^T U + 2aA^TL_vA \]
\[ + 2\beta A^TAZE + \Phi. \]
\[ (12) \]

Using the KKT conditions [41] \( \psi, u_i, u_{ij} = 0, v_i, s_{ij} = 0, \) and \( \phi, z_{ij} = 0 \) and substituting \( L_v = D - W \) and \( L_u = D - W \), the following equations for \( u_i, s_{ij}, \) and \( z_{ij} \) can be obtained:
\[ (XAZS^T + aW^T U - USZ^T A^T AZS^T - aD^TU - \beta UE)_{ii} u_i = 0, \]
\[ (U^TXAZ - U^TUSZ^T A^T AZ - \lambda S)_{s_{ij}} = 0, \]
\[ (A^TX^T U + aA^T W^T AZ - \beta A^TAZE)_{z_{ij}} = 0. \]
\[ (13) \]

Finally, we get the following update rules:
\[ U \leftarrow U \cdot \frac{XAZS^T + aW^TU}{USZ^T A^T AZS^T + aD^TU + \beta UE} \]
\[ S \leftarrow S \cdot \frac{U^TXAZ}{U^TUSZ^T A^T AZ + \lambda S} \]
\[ Z \leftarrow Z \cdot \frac{A^TX^T U + aA^TW^T AZ}{A^TAZ^TU^T U + aA^T D^TAZ + \beta A^TAZE} \]
\[ (14) \]

2.4. Convergence Analysis. We discuss the convergence of CDONMTf under the update rules equations (14)–(16) by the following theorem.

**Theorem 1.** The objective function \( D_{\text{CDONMTf}}(X\|USV^T) \) is nonincreasing under the update rules equations (14)–(16).

In order to prove Theorem 1, the following definition and four Lemmas are introduced first.

**Definition 1** (see [42]). \( G(x, x') \) is an auxiliary function for \( F(x) \) if the conditions
\[ G(x, x') \geq F(x), \]
\[ G(x, x) = F(x), \]
are satisfied.

**Lemma 1** (see [42]). If \( G(x, x') \) is an auxiliary function of \( F(x) \), then \( F(x) \) is nonincreasing under the update:
\[ x' = \arg \min_{x'} G(x, x'). \]
\[ (18) \]

**Proof.**
\[ F(x') \leq G(x'^{+1}, x') \leq G(x', x') = F(x'). \]

For any element \( z_{ij} \) in \( Z \), let \( F_{ij}(z_{ij}) \) be the part about \( Z \) in the objective function equation (9).

Then, we can obtain
\[ F_{ij}(z_{ij}) = (-2A^T X^TU + 2A^T AZU^T U + 2aA^TL_vA + 2\beta A^TAZE)_{z_{ij}}, \]
\[ F_{ij}(z_{ij}) = 2(A^TA)_{ii} (U^T U)_{jj} + 2a(A^TL_vA)_{ii} + 2\beta(A^TA)_{ii} E_{ij}, \]
\[ (20) \]

**Lemma 2.** The function
\[ G(z_{ij}, z_{ij}') = F_{ij}(z_{ij}') + F_{ij}(z_{ij} - z_{ij}') \]
\[ + \frac{(A^T AZU^T U + aA^T D^TAZ + \beta A^TAZE)_{z_{ij}}}{z_{ij}'} \]
\[ : (z_{ij} - z_{ij}')^2, \]
\[ (21) \]

is an auxiliary function for \( F_{ij}(z_{ij}) \).

**Proof.** The Taylor series expansion of \( F_{ij}(z_{ij}) \) is defined as below:
\[ F_{ij}(z_{ij}) = F_{ij}(z_{ij}') + \left[ (A^TA)_{ii} (U^T U)_{jj} + 2a(A^TL_vA)_{ii} + 2\beta(A^TA)_{ii} E_{ij} \right] (z_{ij} - z_{ij}'). \]
\[ (22) \]

Comparing equation (21) with equation (22), it is easy to find that \( G(z_{ij}, z_{ij}') \geq F_{ij}(z_{ij}') \) is equivalent to
\[ \frac{(A^T AZU^T U + aA^T D^TAZ + \beta A^TAZE)_{z_{ij}}}{z_{ij}'} \]
\[ \geq (A^TA)_{ii} (U^T U)_{jj} + a(A^TL_vA)_{ii} + \beta(A^TA)_{ii} E_{ij}, \]
\[ (23) \]
Input: data matrix $X$, constraints matrix $A$, dimension values $p$ and $q$, the neighbor number $k$, parameters $\alpha$, $\beta$, and $\lambda$, and number of iterations $t_{\text{max}}$.  
Output: $U$, $S$, and $Z$.

1. randomly initialize three nonnegative matrices $U$, $S$, and $Z$.
2. use the 0-1 weighted scheme to construct the k-nearest neighbor data graph $W^U$ and feature graph $W^V$.
3. update matrix $D^U_i = \sum_j W_{ij}^U$, $D^S_i = \sum_j W_{ij}^S$, $E = I - T$.
4. for $t = 1, 2, \ldots, t_{\text{max}}$, do
   a. update $U$ as $U \leftarrow U \cdot ((XAZT + \alpha W^U) / (U^T (AZS + \alpha D^ST + \beta E U)))$
   b. update $S$ as $S \leftarrow S \cdot (U^T XAZ / (U^T S^T AZ + \lambda S))$
   c. update $Z$ as $Z \leftarrow Z \cdot ((A^T X^T US + \alpha A^T W^V AZ) / (A^T AZS^T U^T US + \alpha A^T D^V AZ + \beta A^T AZE))$
   End
5. end CDONMTF.

Algorithm 1: CDONMTF.

since

$$
(A^T AZ^T U + \beta A^T AZE)_{ij} = \sum_{l=1}^{r} (A^T Z)_{il} (U^T US)_{ij} + \beta \sum_{l=1}^{r} (A^T Z)_{il} E_{ij}
\geq (A^T Z)_{ij} (U^T US)_{jj} + \beta (A^T Z)_{ij} E_{ij}
\geq \sum_{l=1}^{r} (A^T A)_{il} z_{ij}^l (U^T US)_{ij} + \beta \sum_{l=1}^{r} (A^T A)_{il} z_{ij}^l E_{ij}
\geq z_{ij}^l (A^T A)_{il} (U^T US)_{ij} + \beta z_{ij}^l (A^T A)_{il} E_{ij},
$$

$$
\alpha (A^T D^V AZ)_{ij} = \alpha \sum_{q=1}^{r} (A^T D^V A)_{iq} z_{ij}^q \geq \alpha (A^T D^V A)_{ij} z_{ij}^q
\geq \alpha (A^T (D^V - W^V) A)_{ij} z_{ij}^q = \alpha (A^T L_V A)_{ij} z_{ij}^q.
$$

Thus, equation (23) holds. Then, $G(z_{ij}, z_{ij}^q) \geq F_{ij}(z_{ij})$ and obviously $G(z_{ij}, z_{ij}) = F_{ij}(z_{ij})$. According to the definition of auxiliary function, Lemma 2 is proved.

Denote the part about $S$ in the objective function $D_{\text{CDONMTF}} (X \parallel USV^T)$ as $F_{ij}(s_{ij})$ for any element $s_{ij}$ in $S$, we can obtain

$$
F_{ij}(s_{ij}) = (-2U^T XAZ + 2U^T USZ^T A^T AZ + 2\lambda S)_{ij},
\quad F_{ij}(s_{ij}) = 2(U^T U)^T (Z^T A^T AZ)_{jj} + 2\lambda.
$$

Lemma 3. The function

$$
G(s_{ij}, s_{ij}^l) = F_{ij}(s_{ij}^l) + F_{ij}(s_{ij} - s_{ij}^l)
+ \frac{(U^T USZ^T A^T AZ + \lambda S)_{ij}}{s_{ij}^l} (s_{ij} - s_{ij}^l)^2,
$$

is an auxiliary function for $F_{ij}(s_{ij})$.

Proof. Obviously $G(s, s) = F_{ij}(s)$. According to the definition of auxiliary function, the only thing that needs to be proven is $G(s_{ij}, s_{ij}^l) \geq F_{ij}(s_{ij})$. And $F_{ij}(s_{ij})$ has a Taylor expansion as follows:

$$
F_{ij}(s_{ij}) = F_{ij}(s_{ij}^l) + F_{ij}(s_{ij} - s_{ij}^l) + \frac{(U^T U)^T (Z^T A^T AZ)_{ij}}{(s_{ij} - s_{ij}^l)^2}.
$$

Since

$$
(U^T USZ^T A^T AZ + \lambda S)_{ij} = \sum_{l=1}^{r} (U^T US)_{il} (Z^T A^T AZ)_{ij} + \lambda S_{ij}
\geq (U^T US)_{ij} (Z^T A^T AZ)_{jj} + \lambda S_{ij}
\geq \sum_{l=1}^{r} (U^T US)_{il} s_{ij}^l (Z^T A^T AZ)_{jj} + \lambda s_{ij}^l
\geq s_{ij}^l (U^T U)^T (Z^T A^T AZ)_{jj} + \lambda s_{ij}^l.
$$

Then, $G(s_{ij}, s_{ij}^l) \geq F_{ij}(s_{ij})$ and Lemma 3 holds.

For any element $u_{ij}$ in $U$, the similar results can be obtained as $z_{ij}$ in $Z$. They are
where \( \eta_{ij}, \theta_{ij}, \) and \( \varphi_{ij} \) are the step parameters to control the gradient descent but usually difficult to set. However, due to the special form of partial derivatives, the step parameters can be set automatically by some techniques. Let

\[
\eta_{ij} = - \frac{u_{ij}}{2(UZT^i A^i Z + \alpha D^i U + \beta U E)_{ij}}.
\]

\[
\theta_{ij} = - \frac{s_{ij}}{2(U^TUSZT^i A^i Z + \lambda S)_{ij}},
\]

\[
\varphi_{ij} = - \frac{z_{ij}}{2(A^T AZU^j T + \alpha A^T D^j U + \beta A^T A E)_{ij}}.
\]

We can obtain

\[
\begin{align*}
u_{ij} \gets & u_{ij} + \eta_{ij} \frac{\partial D}{\partial u_{ij}} \\
& = u_{ij} \frac{(XAZ + aW^j U)_{ij}}{(UZT^i A^i AZ + \alpha D^i U + \beta U E)_{ij}} \\
& = s_{ij} \frac{(U^TXAZ)_{ij}}{(U^TUSZT^i A^i AZ + \lambda S)_{ij}} \\
z_{ij} \gets & z_{ij} + \varphi_{ij} \frac{\partial D}{\partial z_{ij}} \\
& = z_{ij} \frac{(A^T X^j U + \alpha A^T W^V AZ)_{ij}}{(A^T AZU^j T + \alpha A^T D^j U + \beta A^T A E)_{ij}}.
\end{align*}
\]

It can be seen that the multiplicative update rules in Equations (14)–(16) are special forms of the step parameters automatically selected by the gradient descent method, so the nonnegativity of \( U, S, \) and \( Z \) can be guaranteed simultaneously.

2.6. Computational Complexity. According to the iterative update rules, it is not difficult to calculate the arithmetic operations of DNMTP and CDONMTF in each iteration. The relative results of NMF and CNMF [27] are also listed here in Table 1. The complexity of each algorithm is \( O(mnp) \). In addition to calculating the hard constraints matrix \( A(V \leftarrow AZ) \), the proposed CDONMTF framework is similar to the corresponding DNMTP structure [35]. The construction of \( k \)-nearest neighbor feature graph and data graph needs \( O(mnt + mn^2) \), and \( W^U \) and \( W^V \) are sparse. Both \( D^U \) and \( D^V \) are diagonal matrices, and \( A \) is a sparse matrix with only \( n \) nonzero elements. Therefore, it can be calculated effectively. When the number of iterations is \( t \), the overall cost for CDONMTF is

\[
O(tmn + mn^2 + mn^3).
\]
3. Experiments and Result Analysis

The performance of the CDONMTF algorithm on some open data sets including 5 UCI machine learning data sets and 7 image data sets is tested. All the experiments are implemented on the Matlab R2014A platform, on an Intel Core i7-8700 (3.2 GHz).

3.1. Comparison of Algorithms and Evaluation Criteria. We compare our method to 8 existing representative methods that are similar, in some respects, to ours. These algorithms include GNMF [18], CNMF [27], DNMTF [35], PNMT [7], NMF2L_{2,0} [22], SODNMF [36], G-NBVD [19], and GSR-NBVD [19].

For graph-based algorithms, GNMF, DNMF, G-NBVD, NMF2L_{2,0}, GSR-NBVD, and our CDONMTF, are weighted with 0 and 1, and SODNMF is weighted with heat kernel. The nearest neighbor number is min\{m, n, 5\} as recommended in [18]. For CNMF, SODNMF, and CDONMTF, 20% of the samples in each data set are randomly selected as labeled samples. The number of iterations for all algorithms are set to 100.

In all experiments, clustering accuracy (AC) [44], normalized mutual information (NMI) [44], and purity (PU) [45] are applied as criteria to evaluate the performance of clustering algorithms.

3.2. Parameter Analysis. The CDONMTF model has two types of parameters: decomposition dimension values \( p \) and \( q \); regularization parameters \( \alpha \), \( \beta \), and \( \lambda \). Although many literature studies have studied how to choose the dimension [32, 46, 47] and regularization parameters [7, 48, 49], there is no unique way to select these parameters. For the choice of \( p \) and \( q \), we use the following methods [19]:

\[
p = \theta_p \times c, \\
q = \theta_q \times c,
\]

where \( \theta_p \) and \( \theta_q \) are empirical, usually depending on the type of data sets, and \( c \) is the number of classes to cluster. In our experiments, \( \theta_p = \theta_q = 3 \) are selected for UCI data sets and \( \theta_p = 3 \) and \( \theta_q = 2 \) for image data sets.

For the regularization parameters \( \alpha \) and \( \beta \), their sensitivity is analyzed on 2 representative data sets, JAFFE and SEMEION. Values as \( \alpha = [0.01, 0.1, 1, 10, 100, 1000] \) and \( \beta = [0.01, 0.05, 0.1, 0.5, 1, 5] \) are tested, while \( \lambda \) is fixed to 1.

Figure 1 displays the clustering results of CDONMTF under different values of \( \alpha \) and \( \beta \). It can be seen that they are more robust for parameter \( \alpha \) in a certain range, but more sensitive for parameter \( \beta \). In the following experiments, we select \( \alpha = 100, \beta = 0.5, \) and \( \lambda = 1 \).

3.3. Experiments on UCI Data. In this section, we test clustering performance of different algorithms on 5 UCI nonnegative data sets (http://archive.ics.uci.edu/ml/index.php). Table 2 outlines the basic information of those data sets. The experimental results listed in Table 3 show that the proposed CDONMTF algorithm achieves the best clustering performance in the view of all 3 indexes: AC, NMI, and PU on three data sets (WDBC, lenses, and Chess), while the clustering results on Zoo and soybean data sets are not satisfying but acceptable.

3.4. Experiments on Image Data. We continue to experiment on different types and sizes of some open image data sets: UMIST (http://images.ee.umist.ac.uk/danny/database.html), JAFFE [50], COIL20 [51], Optdigits (http://archive.ics.uci.edu/ml/machine-learning-databases/optdigits/), MNIST1 (a subset of the first 10k samples) (http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html), and SEMEION (https://archive.ics.uci.edu/ml/machine-learning-databases/semeion/) in this part. Both types and sizes of the data sets are listed in Table 4.

We use a subset of the first \( c \) categories as the collection \( X \) for clustering. Different cluster numbers \( c \) are used to assess the efficiency of the algorithms under various sample scales of \( X \) in the experiments.

Experiment results (Figures 2–4) display the performance of different algorithms using different clustering numbers \( c \). Table 5 shows the average performance of different algorithms under the same data set with different clustering numbers \( c \). The optimal results are marked in bold in the table. Several interesting findings are noted as follows:

(i) SODNMF and the proposed CDONMTF achieve better clustering performance than PNMT, GNMF, DNMTF, NMF2L_{2,0}, G-NBVD, and GSR-NBVD in most cases due to adding the priori label information of the data as an additional constraint

(ii) Compared with CNMF only considering priori label information, the proposed CDONMTF considers the geometric structure information contained in data and feature points and combines with

<table>
<thead>
<tr>
<th>Table 1: Arithmetic operations counts for each iteration.</th>
</tr>
</thead>
<tbody>
<tr>
<td>Floating point operation</td>
</tr>
<tr>
<td>NMF</td>
</tr>
<tr>
<td>Addition (+)</td>
</tr>
<tr>
<td>Multiplication (×)</td>
</tr>
<tr>
<td>Division (+)</td>
</tr>
<tr>
<td>Overall</td>
</tr>
<tr>
<td>CNMF [27]</td>
</tr>
<tr>
<td>(2m + 2n - l + c)(n - l + c)p2 + 2(m + n)p²</td>
</tr>
<tr>
<td>(2m + 2n - l + c)(l - n + c)p2 + 2(m + l)p²</td>
</tr>
<tr>
<td>(m + n - l + c)p</td>
</tr>
<tr>
<td>O(mnp)</td>
</tr>
<tr>
<td>DNMTF [35]</td>
</tr>
<tr>
<td>5mn² + 7mp² + 4n²p² + (m + n)(k + 3)p</td>
</tr>
<tr>
<td>mp² (2p + 1)</td>
</tr>
<tr>
<td>O(mn²)</td>
</tr>
<tr>
<td>CDONMTF</td>
</tr>
<tr>
<td>(4p + 3q + k + 4) + 2m²q + np² + 12n</td>
</tr>
<tr>
<td>+ q(4p + 3q + k + 2) + 2m²q + np² + 12n</td>
</tr>
<tr>
<td>(n - l + c)q + pq</td>
</tr>
<tr>
<td>O(mn²)</td>
</tr>
</tbody>
</table>
Figure 1: Clustering performance of CDONMTF versus the value of parameters. (a) JAFFE. (b) SEMEION. (c) JAFFE. (d) SEMEION. (e) JAFFE. (f) SEMEION.
Table 2: Description of the five UCI data sets.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>No. of features (m)</th>
<th>No. of samples (n)</th>
<th>No. of classes (c)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDBC</td>
<td>30</td>
<td>569</td>
<td>2</td>
</tr>
<tr>
<td>Lenses</td>
<td>4</td>
<td>24</td>
<td>3</td>
</tr>
<tr>
<td>Zoo</td>
<td>16</td>
<td>101</td>
<td>7</td>
</tr>
<tr>
<td>Chess</td>
<td>36</td>
<td>3196</td>
<td>2</td>
</tr>
<tr>
<td>Soybean</td>
<td>35</td>
<td>47</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 3: Comparison of clustering performance on UCI data sets.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>GNMF</th>
<th>CNMF</th>
<th>DNMTF</th>
<th>PNMT</th>
<th>NMF2L(2,0)</th>
<th>SODNMF</th>
<th>G-NBVD</th>
<th>GSR-NBVD</th>
<th>CDONMTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDBC</td>
<td>83.83</td>
<td>79.26</td>
<td>82.59</td>
<td>83.46</td>
<td>83.48</td>
<td>90.33</td>
<td>81.02</td>
<td>69.07</td>
<td>92.79</td>
</tr>
<tr>
<td>Lenses</td>
<td>41.67</td>
<td>56.17</td>
<td>50.13</td>
<td>57.33</td>
<td>41.67</td>
<td>37.51</td>
<td>41.67</td>
<td>45.83</td>
<td>57.83</td>
</tr>
<tr>
<td>Zoo</td>
<td>61.01</td>
<td>67.04</td>
<td>60.93</td>
<td>55.15</td>
<td><strong>84.32</strong></td>
<td>73.39</td>
<td>61.82</td>
<td>62.99</td>
<td>67.19</td>
</tr>
<tr>
<td>Chess</td>
<td>55.16</td>
<td>51.69</td>
<td>54.07</td>
<td>52.07</td>
<td>50.99</td>
<td>60.89</td>
<td>55.23</td>
<td>52.94</td>
<td><strong>77.41</strong></td>
</tr>
<tr>
<td>Soybean</td>
<td>74.47</td>
<td>72.34</td>
<td>78.72</td>
<td>55.28</td>
<td><strong>89.36</strong></td>
<td>81.02</td>
<td>76.61</td>
<td>78.09</td>
<td>87.23</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Accuracy (%)</th>
<th>Normalized mutual information (%)</th>
<th>Purity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>WDBC</td>
<td>83.83</td>
<td>41.67</td>
<td>82.18</td>
</tr>
<tr>
<td>Lenses</td>
<td>41.67</td>
<td>56.17</td>
<td>62.51</td>
</tr>
<tr>
<td>Zoo</td>
<td>61.01</td>
<td>67.04</td>
<td>82.18</td>
</tr>
<tr>
<td>Chess</td>
<td>55.16</td>
<td>51.69</td>
<td>55.16</td>
</tr>
<tr>
<td>Soybean</td>
<td>74.47</td>
<td>72.34</td>
<td>78.72</td>
</tr>
</tbody>
</table>

Table 4: Description of the six image data sets.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>No. of features (m)</th>
<th>No. of samples (n)</th>
<th>No. of classes (c)</th>
<th>Data types</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMIST</td>
<td>10304</td>
<td>575</td>
<td>20</td>
<td>Face image</td>
</tr>
<tr>
<td>JAFFE</td>
<td>4906</td>
<td>213</td>
<td>10</td>
<td>Facial expression</td>
</tr>
<tr>
<td>COIL20</td>
<td>1024</td>
<td>1440</td>
<td>20</td>
<td>Object image</td>
</tr>
<tr>
<td>Optdigit</td>
<td>64</td>
<td>3823</td>
<td>10</td>
<td>Handwritten digit</td>
</tr>
<tr>
<td>MNIST1</td>
<td>784</td>
<td>10000</td>
<td>10</td>
<td>Handwritten digit</td>
</tr>
<tr>
<td>SEMEION</td>
<td>256</td>
<td>1593</td>
<td>10</td>
<td>Handwritten digit</td>
</tr>
</tbody>
</table>

Figure 2: Continued.
Figure 2: Comparisons of clustering AC (%) with different number of classes on six data sets. (a) UMIST. (b) JAFFE. (c) COIL20. (d) Optdigit. (e) MNIST1. (f) SEMEION.

Figure 3: Continued.
Figure 3: Comparisons of clustering NMI (%) with different number of classes on six data sets. (a) UMIST. (b) JAFFE. (c) COIL20. (d) Optdigit. (e) MNIST1. (f) SEMEION.

Figure 4: Continued.
additional biorthogonal constraints, which results in more efficient clustering performance.

(iii) From the view of evaluation indexes (AC, NMI, and PU), the average clustering effect of CDONMTF is almost the best on all data sets except JAFFE, but it is also good enough.

3.5. Experiments on Large-Scale Multiclass Data. PRID (Person Re-ID) 2011 data set [52] includes images extracted from multiple tracks recorded by two different static surveillance cameras. These images contain viewpoint changes and significant differences in lighting, background, and camera characteristics. 80 images are selected for each of 200 people (cam b), 16,000 images in total and each image is reshaped to a 1024-dimensional column vector. Some representative images are illustrated in Figure 5.

<table>
<thead>
<tr>
<th>Data sets</th>
<th>Accuracy (%)</th>
<th>Normalized mutual information (%)</th>
<th>Purity (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMIST</td>
<td>43.47</td>
<td>58.33</td>
<td>50.04</td>
</tr>
<tr>
<td>JAFFE</td>
<td>87.89</td>
<td>86.41</td>
<td>88.66</td>
</tr>
<tr>
<td>COIL20</td>
<td>59.71</td>
<td>36.93</td>
<td>62.15</td>
</tr>
<tr>
<td>Optdigit</td>
<td>89.53</td>
<td>49.44</td>
<td>86.66</td>
</tr>
<tr>
<td>MNIST1</td>
<td>56.52</td>
<td>49.44</td>
<td>50.04</td>
</tr>
<tr>
<td>SEMEION</td>
<td>70.56</td>
<td>36.93</td>
<td>62.15</td>
</tr>
</tbody>
</table>

Table 5: Comparison of average clustering performance on six image data sets.

![Figure 4: Comparisons of clustering PU (%) with different number of classes on six data sets. (a) UMIST. (b) JAFFE. (c) COIL20. (d) Optdigit. (e) MNIST1. (f) SEMEION.](image-url)
CDONMTF shows slightly better performance in the view of the average value of AC, NMI, and PU. In addition, CDONMTF has a higher efficiency than SODNMF on the data sets shown in Table 7.

3.6. Clustering Visualization. In order to visualize the effectiveness of our CDONMTF algorithm, we use IRIS data set (https://archive.ics.uci.edu/ml/datasets/iris) for data analysis. The data set contains a total of 150 samples in 3 classes, and each sample contains 4 features. As shown in Figure 6(a), we use PCA to project sample data into two-dimensional space, display the distribution of the 3 classes by different colors and tag shapes, and randomly select 20% samples from each class as prior labeled samples and marked with red dots.

Table 6: Comparison of clustering performance on PRID data set.

<table>
<thead>
<tr>
<th>Algorithms</th>
<th>GNMF</th>
<th>CNMF</th>
<th>DNMTF</th>
<th>PNMT</th>
<th>NMF2L_{(2,0)}</th>
<th>SODNMF</th>
<th>G-NBVD</th>
<th>GSR-NBVD</th>
<th>CDONMTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>c = 50</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
</tr>
<tr>
<td></td>
<td>60.79</td>
<td>55.68</td>
<td>70.13</td>
<td>14.76</td>
<td>69.31</td>
<td>77.51</td>
<td>70.49</td>
<td>70.52</td>
<td>78.71</td>
</tr>
<tr>
<td></td>
<td>78.31</td>
<td>69.35</td>
<td>86.02</td>
<td>31.25</td>
<td>84.94</td>
<td>90.84</td>
<td>86.48</td>
<td>86.5</td>
<td>91.46</td>
</tr>
<tr>
<td></td>
<td>65.14</td>
<td>58.84</td>
<td>76.21</td>
<td>16.16</td>
<td>75.44</td>
<td>81.38</td>
<td>76.62</td>
<td>76.62</td>
<td>81.99</td>
</tr>
<tr>
<td>c = 80</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
</tr>
<tr>
<td></td>
<td>52.72</td>
<td>49.76</td>
<td>65.14</td>
<td>11.58</td>
<td>63.48</td>
<td>74.74</td>
<td>64.63</td>
<td>64.65</td>
<td>76.25</td>
</tr>
<tr>
<td></td>
<td>74.78</td>
<td>68.29</td>
<td>83.37</td>
<td>31.18</td>
<td>81.61</td>
<td>89.76</td>
<td>83.08</td>
<td>83.07</td>
<td>90.23</td>
</tr>
<tr>
<td></td>
<td>57.38</td>
<td>52.71</td>
<td>71.51</td>
<td>12.81</td>
<td>69.98</td>
<td>78.63</td>
<td>71.05</td>
<td>71.05</td>
<td>79.58</td>
</tr>
<tr>
<td>c = 110</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
</tr>
<tr>
<td></td>
<td>54.71</td>
<td>45.18</td>
<td>61.97</td>
<td>9.8</td>
<td>60.58</td>
<td>75.47</td>
<td>62.16</td>
<td>62.15</td>
<td>75.54</td>
</tr>
<tr>
<td></td>
<td>77.04</td>
<td>66.25</td>
<td>82.09</td>
<td>31.59</td>
<td>80.39</td>
<td>90.71</td>
<td>82.37</td>
<td>82.41</td>
<td>90.33</td>
</tr>
<tr>
<td></td>
<td>59.32</td>
<td>48.35</td>
<td>68.89</td>
<td>10.88</td>
<td>67.61</td>
<td>79.32</td>
<td>69.03</td>
<td>69.03</td>
<td>79.11</td>
</tr>
<tr>
<td>c = 140</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
</tr>
<tr>
<td></td>
<td>53.02</td>
<td>44.35</td>
<td>59.59</td>
<td>9.05</td>
<td>59.42</td>
<td>74.81</td>
<td>59.51</td>
<td>59.71</td>
<td>73.78</td>
</tr>
<tr>
<td></td>
<td>76.71</td>
<td>66.51</td>
<td>80.79</td>
<td>32.51</td>
<td>79.93</td>
<td>90.31</td>
<td>80.91</td>
<td>81.03</td>
<td>88.96</td>
</tr>
<tr>
<td></td>
<td>58.21</td>
<td>47.71</td>
<td>67.42</td>
<td>9.95</td>
<td>66.87</td>
<td>78.89</td>
<td>67.26</td>
<td>67.39</td>
<td>77.57</td>
</tr>
<tr>
<td>c = 170</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
</tr>
<tr>
<td></td>
<td>50.28</td>
<td>44.39</td>
<td>58.52</td>
<td>8.36</td>
<td>59.49</td>
<td>72.69</td>
<td>58.46</td>
<td>58.44</td>
<td>73.08</td>
</tr>
<tr>
<td></td>
<td>75.35</td>
<td>67.92</td>
<td>80.55</td>
<td>32.83</td>
<td>80.42</td>
<td>88.71</td>
<td>80.36</td>
<td>80.39</td>
<td>89.66</td>
</tr>
<tr>
<td></td>
<td>55.33</td>
<td>47.74</td>
<td>66.32</td>
<td>9.19</td>
<td>66.95</td>
<td>76.42</td>
<td>66.23</td>
<td>66.22</td>
<td>77.33</td>
</tr>
<tr>
<td>c = 200</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
</tr>
<tr>
<td></td>
<td>51.31</td>
<td>42.12</td>
<td>58.24</td>
<td>7.73</td>
<td>58.42</td>
<td>71.21</td>
<td>58.47</td>
<td>58.38</td>
<td>73.08</td>
</tr>
<tr>
<td></td>
<td>76.64</td>
<td>66.59</td>
<td>79.97</td>
<td>32.98</td>
<td>79.82</td>
<td>87.02</td>
<td>79.93</td>
<td>79.81</td>
<td>89.65</td>
</tr>
<tr>
<td></td>
<td>56.24</td>
<td>45.43</td>
<td>66.11</td>
<td>8.64</td>
<td>66.06</td>
<td>75.32</td>
<td>66.21</td>
<td>66.11</td>
<td>77.16</td>
</tr>
<tr>
<td>Average</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
<td>AC</td>
<td>NMI</td>
<td>Pu</td>
</tr>
<tr>
<td></td>
<td>53.81</td>
<td>46.91</td>
<td>62.26</td>
<td>10.21</td>
<td>61.79</td>
<td>74.41</td>
<td>62.29</td>
<td>62.29</td>
<td>75.07</td>
</tr>
<tr>
<td></td>
<td>76.47</td>
<td>67.48</td>
<td>82.13</td>
<td>32.06</td>
<td>81.19</td>
<td>89.55</td>
<td>82.19</td>
<td>82.21</td>
<td>90.05</td>
</tr>
<tr>
<td></td>
<td>58.59</td>
<td>50.13</td>
<td>69.41</td>
<td>11.27</td>
<td>68.82</td>
<td>78.33</td>
<td>69.41</td>
<td>69.41</td>
<td>78.79</td>
</tr>
</tbody>
</table>
We use CDONMTF, DNMTF, and CNMF algorithms to cluster the original IRIS data and obtain a clustering label for each sample. The clustering label is marked by its class with the same color and a smaller tag shape. If the clustering label is the same as the originally displayed, then we get a data point with a correct classification (Figures 6(b)–6(d)). For the red marked samples, DNMTF has many clustering errors, but CDONMTF and CNMF cluster correctly using the label information. For unlabeled samples, CDONMTF has the highest accuracy owing to its effective use of priori labels, geometric structures of data and feature manifolds, and independent biorthogonal constraints.

### Table 7: Running time of different algorithms on seven image data sets (seconds).

<table>
<thead>
<tr>
<th>Data sets</th>
<th>GNMF</th>
<th>CNMF</th>
<th>DNMTF</th>
<th>PNMT</th>
<th>NMF2LR(2,0)</th>
<th>SODNMF</th>
<th>G-NBVD</th>
<th>GSR-NBVD</th>
<th>CDONMTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>UMIST</td>
<td>0.75</td>
<td>6.29</td>
<td>1.48</td>
<td>4.81</td>
<td>22.32</td>
<td>81.8</td>
<td>4.41</td>
<td>16.31</td>
<td>9.26</td>
</tr>
<tr>
<td>JAFFE</td>
<td>0.08</td>
<td>0.64</td>
<td>0.19</td>
<td>0.57</td>
<td>2.91</td>
<td>12.06</td>
<td>0.54</td>
<td>1.63</td>
<td>1.02</td>
</tr>
<tr>
<td>COIL20</td>
<td>0.18</td>
<td>2.48</td>
<td>0.34</td>
<td>1.07</td>
<td>9.2</td>
<td>28.47</td>
<td>1.03</td>
<td>4.87</td>
<td>3.78</td>
</tr>
<tr>
<td>Optdigit</td>
<td>0.08</td>
<td>1.75</td>
<td>0.14</td>
<td>0.2</td>
<td>19.55</td>
<td>198.67</td>
<td>0.28</td>
<td>1.08</td>
<td>6.14</td>
</tr>
<tr>
<td>MNIST1</td>
<td>1.45</td>
<td>57.16</td>
<td>2.38</td>
<td>6.57</td>
<td>241.99</td>
<td>2738.62</td>
<td>8.7</td>
<td>23.36</td>
<td>87.99</td>
</tr>
<tr>
<td>SEMEION</td>
<td>0.06</td>
<td>0.8</td>
<td>0.01</td>
<td>0.26</td>
<td>5.06</td>
<td>25.86</td>
<td>0.28</td>
<td>1.41</td>
<td>1.58</td>
</tr>
<tr>
<td>PRID</td>
<td>15.69</td>
<td>223.73</td>
<td>28.11</td>
<td>45.59</td>
<td>672.37</td>
<td>9848.17</td>
<td>105.39</td>
<td>419.94</td>
<td>457.96</td>
</tr>
</tbody>
</table>

Figure 6: IRIS data set clustering illustration based on different algorithms. (a) IRIS. (b) CNMF. (c) DNMTF. (d) CDONMTF.

#### 3.7. Analysis of Convergence.

The update rules used to minimize the CDONMTF objective function are essentially iterative processes. We have proved in Theorem 1 that the iterative rules are convergent. In this section, the convergence rate of these update rules is studied experimentally.

Figure 7 shows the convergence curves of CDONMTF on 6 image data sets. All samples of each data set are decomposed in the experiments. In each subfigure, the $y$–axis is the value of the objective function while the $x$–axis denotes the number of iterations. Although the sizes of the 6 data sets are different, the proposed algorithm usually...
converges within 100 iterations, which indicates that CDONMTF has relatively stable and fast convergence rate.

4. Conclusions

A novel coclustering algorithm, namely, CDONMTF is proposed in this paper. The algorithm effectively utilizes the hidden geometric structure information in the data and feature spaces by constructing the associated data and feature graphs and incorporates the priori label information of the data and independent biorthogonal constraints to enhance the ability of data discrimination in the new representation space. The clustering experiments on multiple UCI machine learning data sets and image data sets show that the proposed algorithm has better performance over several existing clustering methods.

Data Availability

The data used to support the findings of this study are available from the corresponding author upon request.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

This work was supported by the 2018 graduate course construction project of Beijing Forestry University under Grant no. HXKC18006 and the National Natural Science Foundation of China under Grant no. 61571046.

References


Submit your manuscripts at
www.hindawi.com