Neural Process Lett (2015) 42:167–185
DOI 10.1007/s11063-014-9350-0

Non-negative Matrix Factorization with Pairwise Constraints and Graph Laplacian

Yang-Cheng He · Hong-Tao Lu ·
Lei Huang · Xiao-Hua Shi

Published online: 12 April 2014

Abstract Non-negative matrix factorization (NMF) is a very effective method for high dimensional data analysis, which has been widely used in information retrieval, computer vision, and pattern recognition. NMF aims to find two non-negative matrices whose product approximates the original matrix well. It can capture the underlying structure of data in the low dimensional data space using its parts-based representations. However, NMF is actually an unsupervised method without making use of prior information of data. In this paper, we propose a novel pairwise constrained non-negative matrix factorization with graph Laplacian method, which not only utilizes the local structure of the data by graph Laplacian, but also incorporates pairwise constraints generated among all labeled data into NMF framework. More specifically, we expect that data points which have the same class label will have very similar representations in the low dimensional space as much as possible, while data points with different class labels will have dissimilar representations as much as possible. Consequently, all data points are represented with more discriminating power in the lower dimensional space. We compare our approach with other typical methods and experimental results for image clustering show that this novel algorithm achieves the state-of-the-art performance.

Keywords Non-negative matrix factorization · Semi-supervised learning · Pairwise constraints · Graph Laplacian · Clustering

Y.-C. He (✉) · H.-T. Lu · L. Huang · X.-H. Shi
MOE-Microsoft Laboratory for Intelligent Computing and Intelligent Systems, Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai 200240, People's Republic of China
e-mail: h331076268@126.com

H.-T. Lu
e-mail: lu-ht@cs.sjtu.edu.cn

L. Huang
e-mail: tommyflynns@sjtu.edu.cn

X.-H. Shi
e-mail: xhshi@sjtu.edu.cn

1 Introduction

Dimensionality reduction has been widely used as a fundamental tool to analyze the high-dimensional data [1,18,20]. Linear discriminant analysis (LDA) [24], principal component analysis (PCA) [25] are the most popular dimensionality reduction techniques. Some dimensionality reduction techniques such as LDA and PCA can be understood as matrix factorization by using different objective function criteria. Matrix factorization approximately decomposes a matrix as a product of two or more matrices.Among existing matrix decomposition methods, non-negative matrix factorization (NMF) [12] can be used to obtain new representations of data points with non-negativity constraints. That is, it requires that all elements of the decomposed factor matrices are non-negative. These non-negative constraints lead to parts-based representations of the objects because they only allow additive, not subtractive, combinations of the original data points. NMF is a helpful dimensionality reduction method for face recognition [7], document clustering [29], image processing [10] and computer vision [22].

Generally, clustering can be divided into unsupervised clustering and semi-supervised clustering. In unsupervised clustering, we don't need to use any label information to cluster data. In semi-supervised clustering, we need labels of some data points to clustering. Semi-supervised clustering methods need some labeled data that can be user specified or randomly selected from the data points. NMF is an unsupervised learning method. NMF does not use any prior knowledge of data to guide the learning process, nevertheless, there is certain amount of prior knowledge in the real world applications. Using prior knowledge to improve the performance of the algorithms has become one of the hot areas of machine learning. Many machine learning researchers have pointed out that when a small amount of labeled data is used in conjunction with unlabeled data, it can produce encouraging improvement in learning performance [3,6,8,30,35]. However, it is infeasible to label all the data points in the database, because the cost will be highly expensive, whereas obtaining a small amount of labeled data is relatively inexpensive. Under these circumstances, semi-supervised learning algorithms can play a greater performance. NMF has been extended to semi-supervised manner to get better performance [9,16,23,31].

Liu et al. [16] proposed a constrained non-negative matrix factorization (CNMF) approach which took the label information as additional constraints. The main idea of their algorithm is that the data points with the same class label must be strictly mapped to share the same representation in the new parts-based representations space. Thus, the method forces the new parts-based representations to have the consistent label information with the original data. Obviously, this requirement is too strict so that it will weaken the representational ability of the new parts-based representations space for other unlabeled data, because it might assign unlabeled data with totally wrong representations due to its constraints. Wang et al. [23] proposed a penalized matrix factorization (PMF) algorithm, which took the form of pairwise constraints as supervisory information. However, the penalties for violating the must-link constraints are hard to fix. Yang et al. [31] proposed a pairwise constraints guided non-negative matrix factorization (PCNMF), which used the pairwise constraints to guide the clustering process.

Recently, manifold learning method [36,37] has also been incorporated into NMF. Cai et al. [2] had proposed a graph regularized NMF (GNMF) algorithm which encoded the geometrical information of the data space by constructing a nearest neighbor graph to model the local manifold structure.

In our previous work [9], we have proposed a Semi-supervised non-negative matrix factorization (SEMINMF) with graph Laplacian method which incorporated label information and graph Laplacian into NMF. However we can only set the dimensionality of the factorized matrices to the number of clusters in SEMINMF, which may result in bigger reconstruction error between the original matrix and the factorized matrices. Besides, the label information used in SEMINMF can be regarded as hard constraints, it forces the factorized coefficient matrix to have the consistent label information with the cluster indicator matrix of the labeled points, which also may generate bigger reconstruction error.

In this paper, we propose a novel pairwise constrained non-negative matrix decomposition with graph Laplacian (PCGNMF) method. Unlike SEMINMF, PCGNMF does not directly use the class label information to clustering, but utilizes the pairwise constraints generated among all the labeled data to enhance the learning quality. The label information used in SEMINMF can be regarded as hard constraints, while pairwise constraints used in PCGNMF can be regarded as soft constraints. PCGNMF can set the dimensionality of the factorized matrices freely, but in SEMINMF the dimensionality of the factorized matrices must be the same as the number of clusters. With the pairwise constraints, PCGNMF requires that two data points having the same class label should have very similar representations in the new parts-based representations space as much as possible. On the contrary, the data points having different class labels should have quite dissimilar representations in the new parts-based representations space. We do not directly use the pairwise constraints to create the graph Laplacian matrix, because the number of pairwise constraints is small, which can not characterize the local structure of the data adequately. So we incorporate graph Laplacian into NMF, it requires that the nearby points should share the similar representations as much as possible. In this way, we expect that PCGNMF can obtain a more compact and discriminative representation for the data. To achieve this, we carefully design a new NMF objective function incorporating the pairwise constraints information and graph Laplacian into it. Our experimental evaluations show that the proposed approach achieves the state-of-the-art performance.

2 Related Works

Given a set of data points matrix $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m\times n}$, $\mathbf{x}_j$, $j = 1, \ldots, n$, is an m-dimensional non-negative vector, denoting the jth data point. NMF aims to factorize $\mathbf{X}$ into the product of two non-negative matrices $\mathbf{U}$ and $\mathbf{V}$. The product of $\mathbf{U}$ and $\mathbf{V}$ is a good approximation to the original matrix, i.e.,

$$\mathbf{X} \approx \mathbf{U}\mathbf{V}^T \tag{1}$$

In order to obtain the two non-negative matrices, we can quantify the quality of the approximation by using a cost function with some distance metric. For example, if the Euclidean distance between two matrices is used, the problem turns to minimize the following objective function.

$$J = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||^2 = \sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - \sum_{c=1}^{k} u_{ic}v_{jc})^2 \tag{2}$$

where $\|.\|$ is the matrix Frobenius norm denoting the squared sum of all the entries in the matrix. The sizes of the factorized matrices $\mathbf{U}$ and $\mathbf{V}$ are $m \times k$ and $n \times k$, respectively. The dimensionality of $\mathbf{U}$ and $\mathbf{V}$ is k. Usually, k is chosen such that $k \ll \min\{m, n\}$. Each column vector $\mathbf{u}_c$ of matrix $\mathbf{U}$ can be regarded as a basis of the new representations space [4,28], while the jth row vector of matrix $\mathbf{V}$ contains the coefficients of a linear combination of the column vectors of $\mathbf{U}$, this linear combination is used to approximate the jth column vector $\mathbf{x}_j$ of matrix $\mathbf{X}$. NMF can derive the latent characteristic structure space $\mathbf{U}$ using the matrix factorization in the clustering process [13,27,29,33].

When NMF is used to deal with clustering tasks, the dimensionality k of the factorized matrices has multiple choices. We can set k to be the same as or bigger than the number of clusters, even we can set k to be smaller than the number of clusters. When we set k to be the same as the number of clusters, each column of decomposed matrix $\mathbf{U}$ can be regarded as the center of one partition of dataset, each data point can be represented by an additive combination of all column vectors of the decomposed matrix $\mathbf{U}$. Each entry in the jth row of the factorized matrix $\mathbf{V}$ is the projection of the jth data point $\mathbf{x}_j$ of the matrix $\mathbf{X}$ onto corresponding column vector of matrix $\mathbf{U}$. Hence, the cluster membership of each data point can be determined by finding the basis (one column of U) with which the data point has the largest projection value. More specifically, we examine each row of $\mathbf{V}$, and assign data point $\mathbf{x}_j$ to cluster c if $c = \arg\max_{c} v_{jc}$ [29]. Certainly, we can also apply K-means to the coefficient matrix for clustering when k is set to be the same as the number of clusters. But if k is set to be bigger or smaller than the number of clusters, we can only apply K-means to the coefficient matrix for clustering.

Cai et al. [2] had proposed a graph regularized NMF (GNMF) algorithm which incorporated the graph Laplacian into NMF. The objective function of GNMF is defined as:

$$J = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||^2 + \lambda \mathrm{tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \tag{3}$$

where $\mathrm{tr}(\cdot)$ is the trace operator, $\mathbf{L}$ is the graph Laplacian matrix, $\mathbf{L} = \mathbf{D} - \mathbf{W}$, $\mathbf{W}$ is the affinity matrix, its entry w_{jq} denotes the similarity between point $\mathbf{x}_j$ and $\mathbf{x}_q$, $\mathbf{D}$ is a diagonal matrix with its entries defined as $d_{jj} = \sum_{q=1}^{n} w_{jq}$. Due to the graph Laplacian matrix, GNMF can effectively utilize the local structure of the data and obtain a compact representation for the data.

Recently, pairwise constraints have been incorporated into NMF. Yang et al. [31] proposed a PCNMF, which utilized the pairwise constraints to improve the performance of NMF. The objective function of PCNMF is defined as:

$$J = ||\mathbf{X} - \mathbf{U}\mathbf{H}^T||^2 + \lambda \mathrm{tr}(\mathbf{H}^T\mathbf{S}\mathbf{H}) \tag{4}$$

$$A_{ij} = \begin{cases} \alpha & \text{if } \mathbf{x}_i, \mathbf{x}_j (i \neq j) \text{ have the same class label} \\ -(1-\alpha) & \text{if } \mathbf{x}_i, \mathbf{x}_j (i \neq j) \text{ have different class labels} \\ 0 & \text{otherwise} \end{cases}$$

where $\mathbf{S} = \mathbf{D} - \mathbf{A}$, $\mathbf{D}$ is a diagonal matrix with its entries defined as $d_{ii} = \sum_{j=1}^{n} A_{ij}$. The objective function of PCNMF looks like the objective function of GNMF, the main difference is that PCNMF only uses the explicit pairwise constraints to construct the graph Laplacian matrix, while GNMF uses all the data to construct a graph to model the local structure.

Liu et al. [16] proposed a CNMF approach, which utilized the label information to enhance the performance of NMF. The objective function of CNMF is defined as:

$$J = ||\mathbf{X} - \mathbf{U}\mathbf{Z}\mathbf{A}||^2 \tag{5}$$

CNMF incorporates the label information by introducing an auxiliary matrix $\mathbf{A}$. For each data point, if $\mathbf{x}_i$ and $\mathbf{x}_j$ have the same class label, they will have the same representation in the new parts-based representations space.

In our previous work [9], we have proposed a SEMINMF method which incorporated label information and graph Laplacian into NMF. The objective function of SEMINMF is defined as:

$$J = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||^2 + \alpha \mathrm{tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) + \beta||\mathbf{V} - \mathbf{Y}||^2 \tag{6}$$

where $\mathbf{L}$ is the graph Laplacian matrix, $\mathbf{Y}$ is the cluster indicator matrix of the labeled points. The main drawback of SEMINMF is that we can only set k to the number of clusters, which may result in bigger reconstruction error between the original matrix and the factorized matrices.

Yang et al. [32] proposed a non-negative spectral clustering with discriminative regularization algorithm, which imposed non-negative constraints to the cluster indicator matrix.

In Sect. 3, we present a novel PCNMF with graph Laplacian method, which incorporates the pairwise constraints generated among the labeled data and graph Laplacian into NMF. The new objective function for NMF is different from these algorithms.

3 NMF with Pairwise Constraints and Graph Laplacian

3.1 The Objective Function

Given a data set consisting of n data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n] \in \mathbb{R}^{m\times n}$, the label information of the first s data points $\mathbf{x}_t(t \leq s)$ is available and the rest points $\mathbf{x}_r(s < r \leq n)$ are unlabeled. When we have these labeled points, we can obtain the specific pairwise constraints information among them. Suppose the data set $\mathbf{X}$ is going to be divided into k clusters, we randomly select f labeled points from each cluster, the pairwise constraints can be easily generated among the labeled points. More specifically, if any two labeled points have the same class label, we generate a must-link constraint for them. If any two labeled points share different class labels, a cannot-link constraint is generated for them. The number of all must-link pairwise constraints and cannot-link pairwise constraints is $k \times C_f^2$ and $f^2 \times C_k^2$, respectively. C_n^m denotes the number of ways to select m from n objects.

Then we can construct a must-link pairwise constraint symmetric matrix $\mathbf{M} = [m_{pj}] \in \mathbb{R}^{n\times n}$ $(p, j = 1, 2, \ldots, n)$ and a cannot-link pairwise constraint symmetric matrix $\mathbf{C} = [c_{pj}] \in \mathbb{R}^{n\times n}$ $(p, j = 1, 2, \ldots, n)$ with the first s labeled data points on the data set as follows:

$$m_{pj} = \begin{cases} 1 & \text{if } \mathbf{x}_i, \mathbf{x}_j (i \neq j) \text{ have the same class label} \\ 0 & \text{otherwise} \end{cases}$$
$$c_{pj} = \begin{cases} 1 & \text{if } \mathbf{x}_i, \mathbf{x}_j (i \neq j) \text{ have different class labels} \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

With the pairwise constraints, our proposed approach reduces to minimize the following objective function:

$$\begin{aligned} J = &\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{ij} - \sum_{c=1}^{k} u_{ic}v_{jc}\right)^2 + \alpha\sum_{c=1}^{k}\sum_{q=1}^{n}\sum_{j=1}^{n} w_{jq}(v_{jc} - v_{qc})^2 \\ &+\beta\sum_{j=1}^{n}\left(\sum_{p:m_{pj}=1}\sum_{c=1}^{k}\sum_{h=1,h\neq c}^{k} v_{jc}v_{ph} + \sum_{p:c_{pj}=1}\sum_{c=1}^{k} v_{jc}v_{pc}\right) \end{aligned} \tag{8}$$

$$w_{jq} = \begin{cases} \exp(-\frac{||\mathbf{x}_j - \mathbf{x}_q||^2}{\sigma^2}) & \text{if} \quad \mathbf{x}_j \in N_p(\mathbf{x}_q) \quad \text{or} \quad \mathbf{x}_q \in N_p(\mathbf{x}_j) \quad \text{and} \quad j \neq q \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

The Eq. (8) can be rewritten in matrix form using an auxiliary matrix $\mathbf{A} \in \mathbb{R}^{k \times k}$, $\mathbf{A}$ is defined as:

$$\mathbf{A} = \begin{pmatrix} 0 & 1 & ... & 1 \\ 1 & 0 & ... & 1 \\ . & . & & . \\ . & . & & . \\ . & . & & . \\ 1 & 1 & ... & 0 \end{pmatrix}$$

$$J = ||\mathbf{X} - \mathbf{U}\mathbf{V}^T||^2 + \alpha \mathrm{tr}(\mathbf{V}^T \mathbf{L} \mathbf{V}) + \beta[\mathrm{tr}(\mathbf{V}^T \mathbf{M} \mathbf{V} \mathbf{A}) + \mathrm{tr}(\mathbf{V}^T \mathbf{C} \mathbf{V})] \tag{10}$$

Although Eq. (10) is compact, it may not easy to understand how it works. Hence, we analyze it with Eq. (8).

In Eq. (9), $N_p(\mathbf{x}_q)$ denotes the set of the p nearest neighbors of the data point $\mathbf{x}_q$. In Eq. (8), $u_{ic} \geq 0$ and $v_{jc} \geq 0$, $i = 1, 2, \ldots, m$; $q, p, j = 1, 2, \ldots, n$; $c = 1, 2, \ldots, k$. The first term in Eq. (8) corresponds to the cost function of NMF, it denotes the squared sum of the Euclidean distance between $\mathbf{X}$ and $\mathbf{U}\mathbf{V}^T$. The second term is graph Laplacian regularization which is used to capture the local structure of the data, it implies that the nearby points should share the similar representations as much as possible. The third term is the cost function for violation of the pairwise constraints. Specifically, the third term includes two components, one is the cost for violation of the must-link constraints, the other is the cost for violation of the cannot-link constraints. We now analyze how the two components of pairwise constraints work when we set k to be the same as the number of clusters:

1. Suppose point $\mathbf{x}_j$ belongs to the cth cluster, then it will have the largest projection value v_{jc} in the jth row of the matrix $\mathbf{V}$ onto corresponding column vector $\mathbf{u}_c$ of the matrix $\mathbf{U}$. If point $\mathbf{x}_p$ has a must-link constraint with point $\mathbf{x}_j$ ($m_{pj} = 1$), then $\mathbf{x}_p$ also belongs to the cth cluster. We expect that $\mathbf{x}_p$ will also have the largest projection value v_{pc} in the pth row of the matrix $\mathbf{V}$ onto corresponding column vector $\mathbf{u}_c$ of the matrix $\mathbf{U}$. In this case, the product of v_{jc} and v_{pc} is the biggest than any other product of v_{jc} and v_{ph} ($h = 1, \ldots, k; h \neq c$) in the jth row and the pth row of the matrix $\mathbf{V}$. Therefore, v_{pc} should be maximized in the p-th row of the matrix $\mathbf{V}$, this is imposed by minimizing $\sum_{j=1}^{n}(\sum_{p:m_{pj}=1} \sum_{c=1}^{k} \sum_{h=1,h\neq c}^{k} v_{jc}v_{ph})$. When $\sum_{j=1}^{n}(\sum_{p:m_{pj}=1} \sum_{c=1}^{k} \sum_{h=1,h\neq c}^{k} v_{jc}v_{ph})$ is minimized, $v_{ph}(h = 1, \ldots, k; h \neq c)$ will be as smaller as possible, while v_{pc} will be getting bigger as much as possible. Eventually, the point $\mathbf{x}_p$ is assigned to the cth cluster as much as possible, as it has the largest projection value v_{pc} in the pth row of the matrix $\mathbf{V}$ onto corresponding column vector $\mathbf{u}_c$ of the matrix $\mathbf{U}$.
2. When two points $\mathbf{x}_j$ and $\mathbf{x}_p$ have a cannot-link constraint ($c_{pj} = 1$), they must be assigned to different clusters. For example, $\mathbf{x}_j$ is going to be assigned to the cth cluster and it will have the largest projection value v_{jc} in the jth row of the matrix $\mathbf{V}$ onto corresponding column vector $\mathbf{u}_c$ of the matrix $\mathbf{U}$. Then $\mathbf{x}_p$ must be assigned to a different cluster, say, the hth ($h \neq c$) cluster, so it will have the largest projection value v_{ph} in the pth row of the matrix $\mathbf{V}$ onto corresponding column vector $\mathbf{u}_h$ of the matrix $\mathbf{U}$. That is, we expect that the jth row and the pth row of the matrix $\mathbf{V}$ are as orthogonal as possible. This can be imposed by minimizing $\sum_{j=1}^{n}(\sum_{p:c_{pj}=1} \sum_{c=1}^{k} v_{jc}v_{pc})$.

In PCGNMF, we can also set k to be smaller or bigger than the number of clusters. When k is different from the number of clusters, how the two components of pairwise constraints work is similar to the above analysis.

If points $\mathbf{x}_j$ and $\mathbf{x}_p$ have a must-link constraint ($m_{pj} = 1$), they should be assigned into the same cluster, equivalently, it means that $\mathbf{x}_j$ and $\mathbf{x}_p$ will have the very similar representations. In other words, v_{jc} should be almost the same as $v_{pc}(c = 1, \ldots, k)$. If v_{jc} is the largest projection value in the jth row of $\mathbf{V}$, we expect that v_{pc} will also be the largest projection value in the pth row of $\mathbf{V}$ as much as possible. This can be imposed by minimizing $\sum_{j=1}^{n}(\sum_{p:m_{pj}=1}\sum_{c=1}^{k}\sum_{h=1,h\neq c}^{k} v_{jc}v_{ph})$. On the contrary, if $\mathbf{x}_j$ and $\mathbf{x}_p$ have a cannot-link constraint ($c_{pj} = 1$), they should be assigned into different clusters, that is to say, $\mathbf{x}_j$ and $\mathbf{x}_p$ will possess quite dissimilar representations. So the jth row and the pth row of the matrix $\mathbf{V}$ should be as orthogonal as possible. This can be imposed by minimizing $\sum_{j=1}^{n}(\sum_{p:c_{pj}=1}\sum_{c=1}^{k} v_{jc}v_{pc})$.

The trade-off these terms is governed by the positive parameters α, β, which specify the relative importance of the reconstruction error, local geometrical structure and the violation of the pairwise constraints.

3.2 The Algorithm

The objective function J of PCGNMF in Eq. (10) is not convex in both two matrix variables $\mathbf{U}$ and $\mathbf{V}$. Therefore, it is unrealistic to find the global minima of J. In the following, we introduce an iterative updating algorithm which can obtain a local optima for J.

Using the matrix property tr($\mathbf{AB}$)=tr($\mathbf{BA}$) and tr($\mathbf{A}$)=tr($\mathbf{A}^T$), the objective function J can be rewritten as following:

$$\begin{aligned} J &= \mathrm{tr}((\mathbf{X} - \mathbf{U}\mathbf{V}^T)^T(\mathbf{X} - \mathbf{U}\mathbf{V}^T)) + \alpha\mathrm{tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) \\ &\quad + \beta[\mathrm{tr}(\mathbf{V}^T\mathbf{M}\mathbf{V}\mathbf{A}) + \mathrm{tr}(\mathbf{V}^T\mathbf{C}\mathbf{V})] \\ &= \mathrm{tr}(\mathbf{X}^T\mathbf{X}) - 2\mathrm{tr}(\mathbf{X}^T\mathbf{U}\mathbf{V}^T) + \mathrm{tr}(\mathbf{V}\mathbf{U}^T\mathbf{U}\mathbf{V}^T) \\ &\quad + \alpha\mathrm{tr}(\mathbf{V}^T\mathbf{L}\mathbf{V}) + \beta[\mathrm{tr}(\mathbf{V}^T\mathbf{M}\mathbf{V}\mathbf{A}) + \mathrm{tr}(\mathbf{V}^T\mathbf{C}\mathbf{V})] \end{aligned} \tag{11}$$

Let ϕ_{ij} and φ_{ij} be the Lagrange multiplier for constraint $u_{ij} \geq 0$ and $v_{ij} \geq 0$, respectively, and $\mathbf{\Phi} = [\phi_{ij}]$, $\mathbf{\Psi} = [\psi_{ij}]$. The Lagrange function $\mathcal{L}$ is

$$\mathcal{L} = J + \mathrm{tr}(\mathbf{\Phi}\mathbf{U}^T) + \mathrm{tr}(\mathbf{\Psi}\mathbf{V}^T) \tag{12}$$

Let the derivatives of $\mathcal{L}$ with respect to $\mathbf{V}$ and $\mathbf{U}$ vanish, we have:

$$\frac{\partial\mathcal{L}}{\partial\mathbf{V}} = -2\mathbf{X}^T\mathbf{U} + 2\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\alpha(\mathbf{D}\mathbf{V} - \mathbf{W}\mathbf{V}) + \beta(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V}) + \mathbf{\Psi} = 0 \tag{13}$$

$$\frac{\partial\mathcal{L}}{\partial\mathbf{U}} = -2\mathbf{X}\mathbf{V} + 2\mathbf{U}\mathbf{V}^T\mathbf{V} + \mathbf{\Phi} = 0 \tag{14}$$

Using the KKT conditions $\psi_{jc}v_{jc} = 0$ and $\phi_{ic}u_{ic} = 0$, we get the following equations for v_{jc} and u_{ic}:

$$v_{jc} \longleftarrow v_{jc}\frac{2(\mathbf{X}^T\mathbf{U})_{jc} + 2\alpha(\mathbf{W}\mathbf{V})_{jc}}{2(\mathbf{V}\mathbf{U}^T\mathbf{U})_{jc} + 2\alpha(\mathbf{D}\mathbf{V})_{jc} + \beta(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V})_{jc}} \tag{15}$$

$$u_{ic} \longleftarrow u_{ic}\frac{(\mathbf{X}\mathbf{V})_{ic}}{(\mathbf{U}\mathbf{V}^T\mathbf{V})_{ic}} \tag{16}$$

Table 1 Parameters used in complexity analysis

Parameters	Description
n	Number of data points
m	Number of features
k	Number of factors
s	Number of labeled data points
m_n	Number of pairwise must-link constraints
c_n	Number of pairwise cannot-link constraints

3.3 Computational Complexity Analysis

The objective function of PCGNMF is minimized by iteratively updating matrices **U** and **V**. In this section, we will discuss the extra computational cost of our PCGNMF algorithm.

The big O analysis is usually used to express the complexity of the algorithm [15]. However, it may be not precise enough to differentiate the complexity of PCGNMF. Thus, we count the arithmetic operations for PCGNMF algorithm [2,15]. Three arithmetic operations addition, multiplication and division are involved in the updating computation. All these operations are performed on floating-point numbers [15]. Table 1 has described the parameters used in the complexity analysis.

Based on the updating rules, we count the number of operations for each update step in PCGNMF. It is important to note that **M** and **C** are sparse matrices, we use m_n and c_n to denote the number of pairwise must-link constraints and pairwise cannot-link constraints, respectively. Thus, we only need $(m_nk + nk^2)$ flam (a floating point addition and multiplication) to compute **MVA** and c_nk flam to compute **CV**. Moreover, **W** is also a sparse matrix, we only need npk flam to compute **WV** [2]. So PCGNMF needs $(2mnk + m_nk + c_nk + 5nk + npk + 2mk^2 + 3nk^2)$ fladd (a floating point addition), $(2mnk+m_nk+c_nk+npk+3nk+2mk^2+3nk^2)$ flmlt (a floating point multiplication) and $(mk+nk)$ fldiv (a floating point division) in each iteration. Besides the multiplicative updating, PCGNMF needs $O(s^2)$ to construct the constraint matrices **M** and **C**, and PCGNMF also needs $O(n^2m)$ to construct the p-nearest neighbor graph [2].

Suppose the multiplicative updates stop after t iterations, the overall computational complexity for PCGNMF will be $O(tmnk + s^2 + n^2m)$.

4 Experimental Results

In this section, The image clustering tasks are used for the performance evaluations of our proposed PCGNMF algorithm.

4.1 Evaluation Metrics

Two metrics are used to evaluate the clustering performance on each experiment [2,14,29]. Experimental result is evaluated by comparing the cluster label of each sample point with its label provided by the dataset. One metric is the accuracy (AC), which can be used to measure the percentage of correct labels obtained by the algorithm. Given a dataset including n images, let l_i and γ_i be the cluster label and the label provided by the dataset of the ith sample point, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{n} \delta(\gamma_i, map(l_i))}{n} \tag{17}$$

where n denotes the total number of images in the dataset. $\delta(x, y)$ is the delta function that equals one if x = y and equals zero otherwise, and map(l_i) is the mapping function that maps each cluster label l_i to the equivalent label from the dataset. The best mapping can be found by using the Kuhn-Munkres algorithm [17].

The second metric is the normalized mutual information (NMI). In clustering problems, mutual information can measure how similar two clusters are. Given two sets of image clusters C and C', their mutual information metric $MI(C, C')$ is defined as follows:

$$MI(\mathcal{C}, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot \log \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)} \tag{18}$$

where $p(c_i)$, $p(c'_j)$ denote the probabilities that an image arbitrarily selected from the data set belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ denotes the joint probability that this arbitrarily selected image belongs to the cluster c_i as well as c'_j at the same time. MI(C,C') takes values between zero and max(H(C),H(C')), where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It reaches the maximum max($H(C)$, $H(C')$) when the two sets of image clusters are identical and becomes zero when the two sets are completely independent. One important character of $MI(C, C')$ is that the value keeps the same for all kinds of permutations [14]. We use the following normalized metric $NMI(C, C')$ which takes values between zero and one:

$$NMI(\mathcal{C}, \mathcal{C}') = \frac{MI(C, C')}{\max(H(C), H(C'))} \tag{19}$$

4.2 Performance Evaluations and Comparisons

To evaluate how the clustering performance can be improved by our method, we compare our algorithm with other five state-of-the-art algorithms:

1. NMF based clustering [29].
2. Graph regularized Non-negative Matrix Factorization (GNMF) which utilizes the local structure of the data by the graph Laplacian [2].
3. CNMF which takes label information as additional constraints [14].
4. PCNMF which incorporates the pairwise constraints information of the data into NMF [31].
5. SEMINMF with graph Laplacian method which incorporates label information and graph Laplacian into NMF [9].

We conduct the performance evaluations using four image datasets. The descriptions of the four datasets are summarized in Table 2, each dataset contains a certain number of categories of images. The detailed descriptions for each image dataset will be introduced later. Generally, when we use NMF to deal with clustering tasks, we set k to the number of clusters [2,14,29,31]. In some cases, if k is different from the number of clusters, the

Table 2 Descriptions of the four databases

Dataset	Size	Dimensionality	Clusters number
AT&T	400	1024	40
Yale	165	1024	15
AR	1399	2580	100
USPS	9298	256	10

Table 3 Clustering accuracy comparison on the four databases

Methods	Accuracy (%)			
	AT&T	Yale	AR	USPS
NMF	57.4 ± 5.6	45.6 ± 3.3	48.7 ± 6.5	66.3 ± 5.5
GNMF	69.4 ± 4.2	48.9 ± 3.4	47.0 ± 6.8	76.9 ± 7.3
PCNMF	67.5 ± 3.6	49.7 ± 4.3	57.2 ± 7.2	69.8 ± 11.3
CNMF	71.6 ± 4.7	52.7 ± 4.4	60.1 ± 4.8	74.4 ± 6.1
SEMINMF	83.9 ± 2.3	60.4 ± 3.4	**76.9 ± 3.5**	80.4 ± 5.5
PCGNMF	**84.1 ± 3.2**	**64.4 ± 5.2**	74.7 ± 3.1	**87.8 ± 2.0**

Bold values signify the best result

Table 4 Clustering normalized mutual information comparison on the four databases

Methods	Normalized mutual information (%)			
	AT&T	Yale	AR	USPS
NMF	70.5 ± 4.3	44.4 ± 2.9	62.2 ± 6.6	54.3 ± 4.6
GNMF	79.0 ± 2.9	46.0 ± 3.6	59.2 ± 7.0	**76.4 ± 3.6**
PCNMF	78.8 ± 3.0	46.8 ± 3.6	69.8 ± 5.5	57.2 ± 11.9
CNMF	81.2 ± 2.7	52.3 ± 4.0	70.9 ± 3.8	62.0 ± 4.3
SEMINMF	84.5 ± 2.2	51.8 ± 3.2	**76.8 ± 3.7**	62.1 ± 6.2
PCGNMF	**86.2 ± 2.4**	**58.5 ± 4.9**	75.2 ± 2.7	72.7±3.5

Bold values signify the best result

Table 5 The best clustering accuracy and corresponding *k* of each algorithm comparison on each database

Methods	Accuracy (%)			
	AT&T	Yale	AR	USPS
NMF	58.0 ± 5.0$_{12}$	46.3 ± 4.7$_{17}$	55.9 ± 5.7$_{27}$	66.3 ± 5.5$_{6}$
GNMF	69.4 ± 4.2$_{20}$	49.7 ± 4.6$_{12}$	50.9 ± 7.5$_{26}$	81.2 ± 10.5$_{23}$
PCNMF	69.8 ± 3.0$_{21}$	49.7 ± 4.3$_{10}$	64.3 ± 7.4$_{28}$	77.5 ± 8.6$_{15}$
CNMF	72.8 ± 3.5$_{22}$	57.5 ± 5.6$_{14}$	66.7 ± 4.9$_{30}$	78.4 ± 7.1$_{10}$
SEMINMF	83.9 ± 2.3$_{20}$	60.4 ± 3.4$_{10}$	**76.9 ± 3.5**$_{20}$	80.4 ± 5.5$_{6}$
PCGNMF	**84.1 ± 3.2**$_{20}$	**67.4 ± 5.3**$_{13}$	73.8 ± 3.5$_{23}$	**88.3 ± 2.5**$_{8}$

Bold values signify the best result

Table 6 The best clustering normalized mutual information and corresponding *k* of each algorithm comparison on each database

Methods	Normalized mutual information (%)			
	AT&T	Yale	AR	USPS
NMF	70.6 ± 3.7$_{12}$	45.6 ± 3.4$_{17}$	66.6 ± 5.4$_{27}$	54.3 ± 4.6$_{6}$
GNMF	79.0 ± 2.9$_{20}$	45.7 ± 5.0$_{12}$	64.6 ± 5.1$_{26}$	**79.5 ± 5.8**$_{23}$
PCNMF	79.7 ± 1.6$_{21}$	46.8 ± 3.6$_{10}$	75.0 ± 6.8$_{28}$	65.4 ± 4.4$_{15}$
CNMF	81.5 ± 2.4$_{22}$	54.2 ± 5.5$_{14}$	76.2 ± 4.9$_{30}$	63.6 ± 5.5$_{10}$
SEMINMF	84.5 ± 2.2$_{20}$	51.8 ± 3.2$_{10}$	76.8 ± 3.7$_{20}$	62.1 ± 6.2$_{6}$
PCGNMF	**86.2 ± 2.4**$_{20}$	**62.8 ± 3.8**$_{13}$	**78.4 ± 3.0**$_{23}$	73.3 ± 4.5$_{8}$

Bold values signify the best result

performances of the algorithms may be even better. In order to demonstrate this difference, we first set *k* to the number of clusters, Tables 3 and 4 have shown the performance of each algorithm. Then, we report the best performance and corresponding *k* of each algorithm in Tables 5 and 6. On AT&T, Yale, AR and USPS, the number of categories which is used to clustering is 20, 10, 20, 6, respectively. The experiments are carried out as follows:

(1). We conduct ten independent experiments on each dataset. In each experiment, we randomly select twenty subjects for clustering on AT&T and AR databases. On Yale database, we randomly select ten subjects for clustering. On USPS, we randomly select six subjects for clustering in each experiment.

(2). In our experiments, three images are randomly selected from each cluster with labels on AT&T and Yale datasets. On AR database, we randomly select five images from each category to provide the label information. For USPS dataset, we randomly pick up 10 % images from each cluster as the available label information. For PCGNMF and PCNMF, the pairwise constraints are generated among all the labeled data points on each dataset.

(3). In the clustering process, for NMF, GNMF, PCNMF and CNMF, in order to achieve the best performance, fast K-means algorithm [19] is further applied to the new data representation $\mathbf{V}$ for clustering. For PCGNMF and SEMINMF, we use $\mathbf{V}$ to determine the cluster label of each data point when k is set to the number of clusters. That is, we examine each row of $\mathbf{V}$, and assign data point $\mathbf{x}_j$ to cluster c if $c = \arg\max_k v_{jk}$. If k is set to be smaller or bigger than the number of clusters, we can apply fast K-means algorithm to the new data representation $\mathbf{V}$ obtained by PCGNMF for clustering.

The above process is repeated ten times, we calculate the average AC and NMI over the ten tests. For each algorithm, in order to achieve its best results, the parameters are appropriately selected. In GNMF, the regularization parameter λ searches the grid $\{0.01, 0.1, 1, 10, 100, 500, 1000\}$. For PCNMF, λ searches the grid $\{0.01, 0.1, 1, 10, 100\}$, α searches the grid $\{0.8, 0.85, 0.9, 0.95, 0.99\}$.For SEMINMF, the regularization parameter α is set by searching the grid $\{200, 260, 320, 380, 440, 500, 560, 620\}$, β searches the grid $\{6, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$. For FCGNMF, α searches the grid $\{0.01, 0.1, 1, 10\}$, β is set by searching the grid $\{1, 10, 20, 30, 60, 100\}$, the number of the nearest neighbors p searches the grid $\{3, 4, 5, 6, 7, 8, 9, 10\}$, in our all experiments, we simply fix $\alpha = 0.1$, $\beta = 20$, $p = 3$. For GNMF and PCGNMF, the parameter σ^2 is set to 1 on each database. For SEMINMF, the parameter σ^2 is set to 1 on AT&T, Yale and AR, on USPS, σ^2 is set to 0.1.

4.3 Data Sets

4.3.1 AT&T Dataset

The AT&T[1] dataset contains 400 images of 40 distinct subjects. Each subject has 10 different images. For some subjects, the images were taken at different times, varying the lighting, facial expressions (open / closed eyes, smiling / not smiling) and facial details (glasses / no glasses). All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position (with tolerance for some side movement). In all the experiments, the original images are normalized in scale and orientation such that the two eyes are aligned at the same position. Then, the facial areas are cropped into the final images for clustering. The size of each cropped image is 32×32 pixels, with 256 gray levels per pixel. Thus, each image can be represented by a 1,024-dimensional vector [14].

4.3.2 Yale Dataset

The Yale Face[2] database contains 165 grayscale images in GIF format of 15 individuals. There are 11 images per subject, one per different facial expression or configuration: center-light, w/glasses, happy, left-light, w/no glasses, normal, right-light, sad, sleepy, surprised,

[1] http://www.face-rec.org/databases/.

[2] http://www.face-rec.org/databases/.

Table 7 Reconstruction errors of PCGNMF and SEMINMF on AR database

Database	$\|\|\mathbf{X} - \mathbf{U}\mathbf{V}^T\|\|^2$			
	SEMINMF	PCGNMF		
	k =20	k =20	k =25	k =29
AR	26.1	10.6	9.4	8.7

and wink. Preprocessing for this dataset has done the same as the AT&T dataset. Each image can also be represented by a 1,024-dimensional vector.

4.3.3 AR Dataset

The AR database consists of over 4,000 frontal images for 126 individuals. We select a subset (with only illumination and expression changes) containing 50 male subjects and 50 female subjects, the total images is 1,399 [26,34].

4.3.4 USPS Dataset

The USPS[3] handwritten digit database contains 10 objects. We select a popular subset containing 9298 16×16 handwritten digit images in total.

When k is set to the number of clusters, Tables 3 and 4 show the detailed clustering accuracy, normalized mutual information and standard deviations on the four datasets. On AT&T, we can see that SEMINMF gets the second best performance, PCGNMF achieves 0.2 % improvement in accuracy and 1.7 % improvement in normalized mutual information over SEMINMF on average. On Yale, SEMINMF obtains the second best result for accuracy, CNMF gets the second best performance in normalized mutual information, PCGNMF improves 4 % in accuracy and 6.2 % in normalized mutual information over SEMINMF and CNMF. On AR, SEMINMF is the best algorithm, PCGNMF gets the second best performance. On USPS, we can see that the local structure of the data is particularly important, GNMF even obtains the best result for normalized mutual information with graph Laplacian only. PCGNMF gets the best result for accuracy.

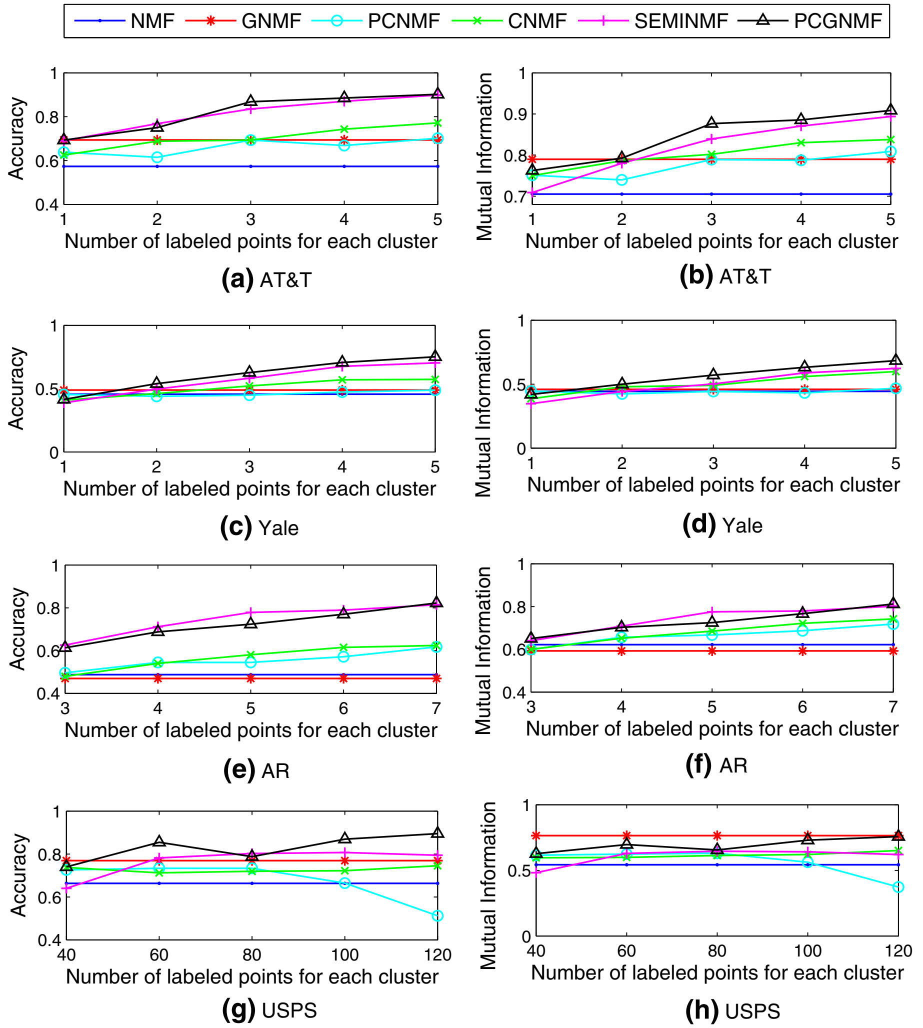
Tables 5 and 6 (the subscripts in the tables denote the dimensionality k of the factorized matrices.) show the best performance and corresponding k of each algorithm on all the databases. Note that in SEMINMF, we can only set k to the number of clusters, the results of SEMINMF are the same as in Tables 3 and 4. In order to compare with others algorithms, we list the results of SEMINMF again. On AT&T, NMF achieves the best performance when the dimensionality k of the factorized matrices is 12, GNMF and PCGNMF obtain the best performances when k is the same as the number of clusters, PCNMF and CNMF have slight improvements in performances when k is 21 and 22, respectively. On Yale, when k is bigger than the number of clusters, NMF, GNMF, CNMF, and PCGNMF obtain better performances. On AR, PCGNMF gets the best result for normalized mutual information when k is 23, but when k is 20 the normalized mutual information of PCGNMF is worse than that of SEMINMF. k is limited to the number of clusters, which is the main drawback of SEMINMF. On USPS, the performances of GNMF, PCNMF, CNMF and PCGNMF have been improved when k is bigger than 6, PCGNMF still gets the best performance in accuracy.

Tables 7 and 8 show the reconstruction errors of PCGNMF and SEMINMF on AR and USPS databases. In SEMINMF, we can only set k to the number of clusters, which may

[3] http://www.cad.zju.edu.cn/home/dengcai/Data/MLData.html.

Table 8 Reconstruction errors of PCGNMF and SEMINMF on USPS database

Database	$\|\|\mathbf{X}-\mathbf{U}\mathbf{V}^T\|\|^2$			
	SEMINMF	PCGNMF		
	k =6	k =6	k =20	k =30
USPS	517.0	498.0	367.2	325.5



(a) AT&T

(b) AT&T

(c) Yale

(d) Yale

(e) AR

(f) AR

(g) USPS

(h) USPS

Fig. 1 When the number of labeled points varies, the performances of the algorithms on each database

result in bigger reconstruction error between the original matrix and the factorized matrices. Besides, the label information used in SEMINMF can be regarded as hard constraints, SEMINMF forces the factorized coefficient matrix to fit the cluster indicator matrix of the labeled points, which is too strict so that it may also lead to bigger reconstruction error. From Tables 7 and 8, we can see that when k is the same as the number of clusters, the reconstruction error

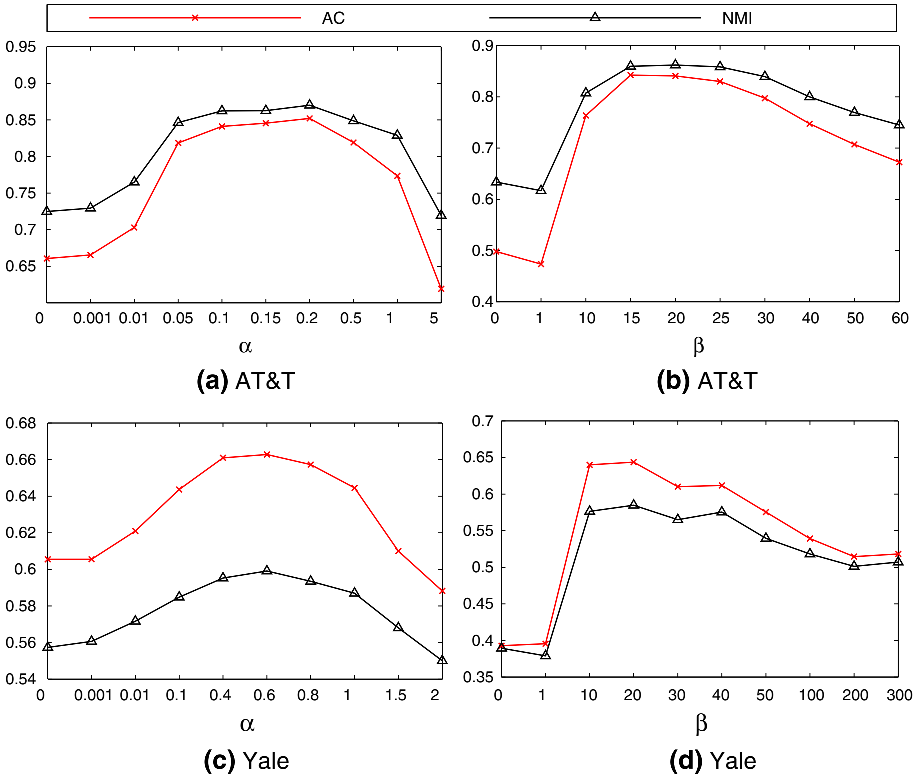


Fig. 2 The performance of PCGNMF versus **a**, **c** α with β fixed, **b**, **d** β with α fixed on AT&T and Yale, respectively

of PCGNMF is smaller than that of SEMINMF. When k becomes large, the reconstruction error of PCGNMF becomes smaller, so the product of $\mathbf{U}$ and $\mathbf{V}$ will be a better approximation of $\mathbf{X}$.

4.4 Parameters Selection

Our PCGNMF algorithm has three main parameters: the number of labeled points, the regularization parameters α and β. In this section, we illustrate the effect of the parameters on performance.

Figure 1 shows how the performances of semi-supervised algorithms vary with the increase of labeled points. On AT&T, as can be seen, when the number of the labeled points increases, the performances of PCNMF, CNMF, SEMINMF and PCGNMF have been improved significantly. On Yale, when the number of labeled points increases, PCGNMF can make use of the label information to enhance the performance and obtain the best results. The performance of PCNMF is not improved significantly when the number of the labeled points increases. On AR, when the number of labeled points is 7 for each cluster, PCGNMF will be as good as SEMINMF. On USPS, the performance of CNMF is not improved significantly when the number of labeled points increases, the performance of PCNMF even degrades. When the number of labeled points is 120, the normalized mutual information of PCGNMF is competitive with that of GNMF.

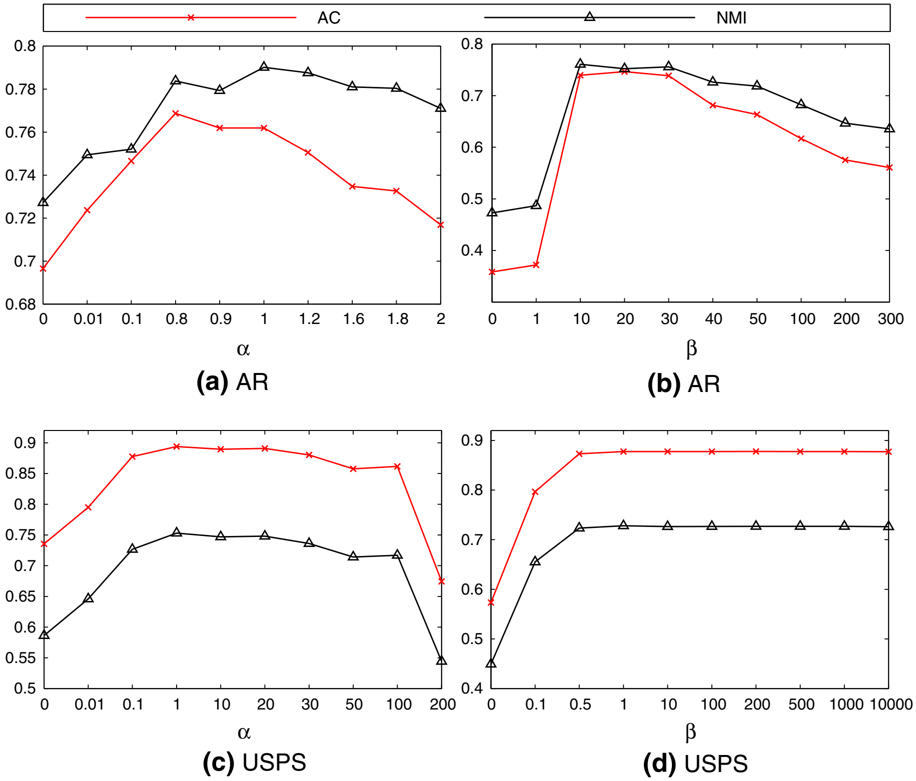


Fig. 3 The performance of PCGNMF versus **a**, **c** α with β fixed, **b**, **d** β with α fixed on AR and USPS, respectively

To study how the two regularization parameters α and β affect the image clustering performance, we carry out some experiments on the parameters sensitivity with α and β varying respectively. Figure 2a shows the performance of PCGNMF varies with α when β is fixed on AT&T. We can see that when α varies from 0 to 0.1, the performance of PCGNMF has been improved. When α varies from 0.1 to 0.2, PCGNMF consistently achieves good and stable performance. When α is greater than 0.2, the performance obviously declines. Figure 2b shows the performance varies with β when α is fixed, from which we can see that when β varies from 1 to 15, the performance has been improved significantly, when β is greater than 25 and 20, the accuracy and the normalized mutual information of PCGNMF will drop, respectively. Figure 3a shows the performance of PCGNMF varies with α when β is fixed on USPS. It can be seen that when α varies from 0 to 1, the performance has been improved significantly, PCGNMF consistently achieves good and stable performance when α varies from 1 to 20. When α is fixed, Fig. 3b shows the performance varies with β, we can see that when β varies from 0 to 0.5, the performance has been improved significantly, PCGNMF consistently achieves good and very stable performance when α varies from 0.5 to 10,000. From Fig. 2 to Fig. 3, we can see that the local structure and the pairwise constraints of the data are all important, with combination of the graph Laplacian and the pairwise constraints, PCGNMF obtains a more compact and discriminative representation for the data and so it can achieve good performance.

5 Conclusions

In order to enhance the performance of NMF, label information and pairwise constraints have been incorporated into NMF. However, some previous existing methods can not make full use of the pairwise constraints and label information to improve the performance of NMF. The CNMF proposed by Liu et al. [16] did not consider that data points with different class labels should have dissimilar representations. The PCNMF proposed by Yang et al. [31] did not consider the local structure of the data. Our previous work SEMINMF [9] incorporated label information as hard constraints and graph Laplacian into NMF. However, SEMINMF can only set the dimensionality of the factorized matrices to the number of clusters, which is the main drawback of SEMINMF.

In this paper, the proposed PCGNMF algorithm takes pairwise constraints of the labeled data points and the local structure of the data with graph Laplacian into account. In PCGNMF, we can set the dimensionality of the factorized matrices freely, so the model is more flexible.

Our experimental evaluations for image clustering tasks show that the proposed algorithm is effective and achieves the state-of-the-art performance. Compared with SEMINMF, the reconstruction error of PCGNMF is smaller than that of SEMINMF, which means that the product of the factorized matrices obtained by PCGNMF will be a better approximation of original data matrix.

Acknowledgments This work was supported in part by NSFC (no. 61272247), the National Basic Research Program of China (973 program) under Grant 2009CB320901, the National High Technology Research and Development Program of China (863 program) under Grant 2008AA02Z310, the National Natural Science Foundation of China under Grant 60873133, arts and Science Cross Special Fund of Shanghai Jiao Tong University under Grant 13JCY14.

6 Appendix

In this section, we prove the convergence of PCGNMF. We begin with the following theorem regarding the iterative updating rules in Eqs. (15) and (16).

Theorem 1 *The objective function J is nonincreasing under the iterative updating rules in Eqs. (15) and (16). The objective function is invariant under these updates if and only if* $\mathbf{U}$ *and* $\mathbf{V}$ *are at a stationary point.*

Theorem 1 guarantees that these iterative updating rules of $\mathbf{U}$ and $\mathbf{V}$ in Eqs. (15) and (16) can converge on a stationary point and hence final solution will be a local optimum. To prove Theorem 1, we have to show that J is nonincreasing under the iterative updating rules in Eqs. (15) and (16). Since the second term and the third term of J are only related to $\mathbf{V}$, and the iterative updating rule (16) is exactly the same as update formula for $\mathbf{U}$ in the NMF. The convergence proof of NMF has shown that J is nonincreasing under the iterative updating rule in Eq. (16) [11]. So, we only need to prove that J is nonincreasing under the iterative updating rule in Eq. (15). Firstly, we make use of a similar auxiliary function which has been used in the Expectation-Maximization algorithm [5,21].

Definition $G(v, v')$ is an auxiliary function for $F(v)$ if the conditions $G(v, v') \geq F(v)$, $G(v, v) = F(v)$ are satisfied.

We have the following lemma regarding the very useful auxiliary function, which will be helpful to prove the convergence of the objective function.

Lemma 1 *If G is an auxiliary function of F, then F is nonincreasing under the update*

$$v^{(t+1)} = \arg\min_{v} G(v, v^t) \tag{20}$$

Proof $F(v^{(t+1)}) \leq G(v^{(t+1)}, v^t) \leq G(v^t, v^t) = F(v^t)$

Now, we will prove that the iterative updating rule for **V** in Eq. (15) is exactly the update rule in Eq. (20) with an appropriate auxiliary function. For any entry v_{ab} in **V**, we use $F_{v_{ab}}$ to denote the part of J only relevant to v_{ab}. It is easy to check that

$$\begin{aligned} F'_{v_{ab}} &= (\frac{\partial J}{\partial \mathbf{V}})_{ab} = -2(\mathbf{X}^T\mathbf{U})_{ab} + 2(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + 2\alpha(\mathbf{D}\mathbf{V})_{ab} \\ &\quad -2\alpha(\mathbf{W}\mathbf{V})_{ab} + \beta(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V})_{ab} \end{aligned} \tag{21}$$

$$F''_{v_{ab}} = 2(\mathbf{U}^T\mathbf{U})_{bb} + 2\alpha\mathbf{D}_{aa} - 2\alpha\mathbf{W}_{aa} + \beta\mathbf{C}_{aa} \tag{22}$$

Where F', F'' are the first and second order derivative with respect to **V**, respectively. □

Lemma 2 *The function*

$$\begin{aligned} G(v, v_{ab}^{(t)}) &= F_{v_{ab}}(v_{ab}^{(t)}) + F'_{v_{ab}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) \\ &\quad + \frac{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + \alpha(\mathbf{D}\mathbf{V})_{ab} + \frac{\beta}{2}(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V})_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2 \end{aligned} \tag{23}$$

is an auxiliary function for $F_{v_{ab}}$, *and it is the part of* J *related* v_{ab}.

Proof Since $G(v, v) = F_{v_{ab}}(v)$ is explicit, we only have to show that $G(v, v_{ab}^{(t)}) \geq F_{v_{ab}}(v)$. In order to achieve that, we can compare the Taylor series expansion of $F_{v_{ab}}(v)$ with the auxiliary function $G(v, v_{ab}^{(t)})$.

$$\begin{aligned} F_{v_{ab}}(v) &= F_{v_{ab}}(v_{ab}^{(t)}) + F'_{v_{ab}}(v_{ab}^{(t)})(v - v_{ab}^{(t)}) \\ &\quad + \left[(\mathbf{U}^T\mathbf{U})_{bb} + \alpha\mathbf{D}_{aa} - \alpha\mathbf{W}_{aa} + \frac{\beta}{2}\mathbf{C}_{aa}\right](v - v_{ab}^{(t)})^2 \end{aligned} \tag{24}$$

Clearly, showing $G(v, v_{ab}^{(t)}) \geq F_{v_{ab}}(v)$ is equivalent to prove that

$$\begin{aligned} \frac{(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + \alpha(\mathbf{D}\mathbf{V})_{ab} + \frac{\beta}{2}(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V})_{ab}}{v_{ab}^{(t)}} &\geq (\mathbf{U}^T\mathbf{U})_{bb} + \alpha\mathbf{D}_{aa} \\ &\quad -\alpha\mathbf{W}_{aa} + \frac{\beta}{2}\mathbf{C}_{aa} \end{aligned} \tag{25}$$

In order to prove above inequality holds, we have

$$(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} = \sum_{c=1}^{k} v_{ac}^{(t)}(\mathbf{U}^T\mathbf{U})_{cb} \geq v_{ab}^{(t)}(\mathbf{U}^T\mathbf{U})_{bb} \tag{26}$$

and

$$\alpha(\mathbf{D}\mathbf{V})_{ab} = \alpha\sum_{j=1}^{n} \mathbf{D}_{aj}v_{jb}^{(t)} \geq \alpha\mathbf{D}_{aa}v_{ab}^{(t)} \tag{27}$$

$$\frac{\beta}{2}(\mathbf{C}\mathbf{V})_{ab} = \frac{\beta}{2}\sum_{j=1}^{n} \mathbf{C}_{aj}v_{jb}^{(t)} \geq \frac{\beta}{2}v_{ab}^{(t)}\mathbf{C}_{aa} \tag{28}$$

Therefore, the inequality $G(v, v_{ab}^{(t)}) \geq F_{v_{ab}}(v)$ holds. □

Now, we can show the convergence of Theorem 1 for **V**:

Proof of Theorem 1 we can replace $G(v, v_{ab}^{(t)})$ in Eq. (20) by Eq. (23) to obtain the update rule which is exactly the same as the iterative updating rule for **V**.

$$\begin{aligned} v_{ab}^{(t+1)} &= v_{ab}^{(t)} \frac{[2\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\alpha\mathbf{D}\mathbf{V} + \beta(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V})]_{ab} - F'_{v_{ab}}(v_{ab}^{(t)})}{[2\mathbf{V}\mathbf{U}^T\mathbf{U} + 2\alpha\mathbf{D}\mathbf{V} + \beta(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V})]_{ab}} \\ &= v_{ab}^{(t)} \frac{2(\mathbf{X}^T\mathbf{U})_{ab} + 2\alpha(\mathbf{W}\mathbf{V})_{ab}}{2(\mathbf{V}\mathbf{U}^T\mathbf{U})_{ab} + 2\alpha(\mathbf{D}\mathbf{V})_{ab} + \beta(\mathbf{M}\mathbf{V}\mathbf{A} + \mathbf{C}\mathbf{V})_{ab}} \end{aligned} \tag{29}$$

Since Eq. (23) is an auxiliary function, $F_{v_{ab}}$ is nonincreasing under this updating rule with Lemma 2. □

References

1. Belkin M, Niyogi P (2003) Laplacian eigenmaps for dimensionality reduction and data representation. Neural Comput 15(6):1373–1396
2. Cai D, He X, Han J, Huang T (2011) Graph regularized nonnegative matrix factorization for data representation. IEEE Trans Pattern Anal Mach Intell 33(8):1548–1560
3. Chapelle O, Schölkopf B, Zien A et al (2006) Semi-supervised learning, vol 2. MIT Press, Cambridge
4. Das Gupta M, Xiao J (2011) Non-negative matrix factorization as a feature selection tool for maximum margin classifiers. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2841–2848
5. Dempster A, Laird N, Rubin D (1977) Maximum likelihood from incomplete data via the em algorithm. J R Stat Soc Ser B (Methodological) 39:1–38
6. Grira N, Crucianu M, Boujemaa N (2005) Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration. In: The 14th IEEE International Conference on Fuzzy Systems, 2005. FUZZ'05. IEEE, pp 867–872
7. Guillamet D, Vitria J (2002) Classifying faces with nonnegative matrix factorization. In: Proceedings of 5th Catalan Conference for Artificial Intelligence
8. He R, Zheng W, Hu B, Kong X (2011) Nonnegative sparse coding for discriminative semi-supervised learning. In: 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE, pp 2849–2856
9. He Y, Lu H, Xie S (2013) Semi-supervised non-negative matrix factorization for image clustering with graph laplacian. Multim Tools Appl. doi:10.1007/s11042-013-1465-1
10. Kim J, Park H (2008) Sparse nonnegative matrix factorization for clustering. CSE Technical Reports, Georgia Institute of Technology, pp 1–16
11. Lee D, Seung H (2001) Algorithms for non-negative matrix factorization. Adv Neural Inf Process Syst 13:556–562
12. Lee D, Seung H et al (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401(6755):788–791
13. Li T, Ding C (2006) The relationships among various nonnegative matrix factorization methods for clustering. In: 6th International Conference on Data Mining, 2006. ICDM'06. IEEE, pp 362–371
14. Liu H, Wu Z (2010) Non-negative matrix factorization with constraints. In: 24th AAAI Conference on Artificial Intelligence.
15. Liu H, Wu Z, Cai D, Huang T (2011) Constrained non-negative matrix factorization for image representation. IEEE Trans Pattern Anal Mach Intell 99:1–1
16. Liu H, Wu Z, Li X, Cai D, Huang TS (2012) Constrained nonnegative matrix factorization for image representation. IEEE Trans Pattern Anal Mach Intell 34(7):1299–1311
17. Lovász L, Plummer M (1986) Matching theory. Elsevier Science Ltd., Amsterdam, p 121
18. Niyogi X (2004) Locality preserving projections. In: Advances in neural information processing systems 16: proceedings of the 2003 conference, vol. 16. The MIT Press, p 153
19. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: IEEE Conference on Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE, pp 1–8

20. Roweis S, Saul L (2000) Nonlinear dimensionality reduction by locally linear embedding. Science 290(5500):2323
21. Saul L, Pereira F (1997) Aggregate and mixed-order markov models for statistical language processing. In: Proceedings of the second conference on empirical methods in natural language processing. Association for Computational Linguistics, Somerset, New Jersey, pp 81–89
22. Shashua A, Hazan T (2005) Non-negative tensor factorization with applications to statistics and computer vision. In: Proceedings of the 22nd international conference on Machine learning. ACM, pp 792–799
23. Wang F, Li T, Zhang C (2008) Semi-supervised clustering via matrix factorization. In: Proceedings of The 8th SIAM Conference on Data Mining
24. Welling M (2005) Fisher linear discriminant analysis. Technical report, vol 3. Department of Computer Science, University of Toronto
25. Wold S, Esbensen K, Geladi P (1987) Principal component analysis. Chemom Intell Lab Syst 2(1–3):37–52
26. Wright J, Yang A, Ganesh A, Sastry S, Ma Y (2009) Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell 31(2):210–227
27. Wu M, Scholkopf B (2007) A local learning approach for clustering. Adv Neural Inf Process Syst 19:1529
28. Xu W, Gong Y (2004) Document clustering by concept factorization. In: Proceedings of the 27th annual international ACM SIGIR conference on research and development in information retrieval. ACM, pp 202–209
29. Xu W, Liu X, Gong Y (2003) Document clustering based on non-negative matrix factorization. In: Proceedings of the 26th annual international ACM SIGIR conference on research and development in informaion retrieval. ACM, pp 267–273
30. Yang L, Jin R, Sukthankar R (2008) Semi-supervised learning with weakly-related unlabeled data: Towards better text categorization. In: 22nd annual conference on neural information processing systems, Citeseer.
31. Yang Y, Hu B (2007) Pairwise constraints-guided non-negative matrix factorization for document clustering. In: IEEE/WIC/ACM international conference on web intelligence. IEEE, pp 250–256
32. Yang Y, Shen HT, Nie F, Ji R, Zhou X (2011) Nonnegative spectral clustering with discriminative regularization. In: AAAI
33. Ye J, Zhao Z, Wu M (2007) Discriminative k-means for clustering. Adv Neural Inf Process Syst 20:1649–1656
34. Zhang L, Yang M, Feng X (2011) Sparse representation or collaborative representation: Which helps face recognition? In: 2011 IEEE international conference on Computer Vision (ICCV). IEEE, pp 471–478
35. Zhang Y, Yeung D (2008) Semi-supervised discriminant analysis using robust path-based similarity. In: IEEE conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE, pp 1–8
36. Zhang Z, Wang J, Zha H (2005) Adaptive manifold learning. IEEE Trans Pattern Anal Mach Intell 99:1–1
37. Zhang Z, Zha H, Zhang M (2008) Spectral methods for semi-supervised manifold learning. In: IEEE conference on Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE, pp 1–6