GRAPH REGULARIZED NON-NEGATIVE LOCAL COORDINATE FACTORIZATION WITH PAIRWISE CONSTRAINTS FOR IMAGE REPRESENTATION

Yangcheng He, Hongtao Lu, Baoliang Lu

Key Laboratory of Shanghai Education Commission for Intelligent Interaction and Cognitive Engineering
Department of Computer Science and Engineering, Shanghai Jiao Tong University, P.R.China
h331076268@126.com,htlu@sjtu.edu.cn,bllu@sjtu.edu.cn

ABSTRACT

Chen *et al.* proposed a non-negative local coordinate factorization algorithm for feature extraction (NLCF) [1], which incorporated the local coordinate constraint into non-negative matrix factorization (NMF). However, NLCF is actually a unsupervised method without making use of prior information of problems in hand. In this paper, we propose a novel graph regularized non-negative local coordinate factorization with pairwise constraints algorithm (PCGNLCF) for image representation. PCGNLCF incorporates pairwise constraints and graph Laplacian into NLCF. More specifically, we expect that data points having pairwise must-link constraints will have the similar coordinates as much as possible, while data points with pairwise cannot-link constraints will have distinct coordinates as much as possible. Experimental results show the effectiveness of our proposed method in comparison to the state-of-the-art algorithms on several real-world applications.

Index Terms— Non-negative, Local Coordinate Coding, Pairwise Constraint, Semi-supervised Learning, Clustering, Sparse Learning

1. INTRODUCTION

Non-negative matrix factorization (NMF) [2] is a useful method for feature extraction, which has been widely applied in computer vision, pattern recognition and data mining. NMF requires that all elements of the decomposed matrix factors are non-negative. These non-negative constraints lead to parts-based representations of the objects. Recently, researchers have pointed out that sparse and parts-based representations can be more robust than non-sparse, global representations [1, 3]. However, NMF does not always obtain sparse representations [1].

To address this problem, Chen *et al.* recently proposed a non-negative local coordinate factorization (NLCF) algorithm [1], which added a local coordinate constraint into NMF objective function. NLCF forces the learned basis vectors to be close to the original data points as much as possible, so that it can naturally leads to sparse representations for the data, because each data point will be represented by a linear combination of only a few nearby basis vectors [1]. However, NLCF can not guarantee that data points with the same class label will have the similar coordinates as much as possible, while data points possessing different class labels will have distinct coordinates as much as possible.

Many machine learning researchers have pointed out that when a small amount of labeled data is used in conjunction with unlabeled data, the performances of learning algorithms can be greatly improved [4, 5, 6]. In order to enhance the performance of NMF, label information has been incorporated into NMF [7, 8]. In this paper, we incorporate pairwise constraints into NLCF to improve its performance.

2. RELATED WORK

Given a set of data points $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots, \mathbf{x}_n] \in \mathbf{R}^{m \times n}$, $\mathbf{x}_j, j = 1, \cdots, n$, is an *m*-dimensional non-negative vector, denoting the j-th 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 $\mathbf{X}$ by minimizing the following objective function:

$$J = \|\mathbf{X} - \mathbf{U}\mathbf{V}\|^2 \tag{1}$$

where $\|.\|$ is the matrix Frobenius norm, i.e., the squared sum of all the entries in the matrix. The dimensions of the factorized matrices $\mathbf{U}$ and $\mathbf{V}$ are $m \times k$ and $k \times n$, respectively. Usually, k is chosen such that $k \ll \min\{m, n\}$. Each column of the decomposed matrix $\mathbf{U}$ can be regarded as an anchor point [1]. The j-th column vector of the matrix $\mathbf{V}$ contains the coefficients of a linear combination of the anchor points,

Before introduction of NLCF algorithm, we first introduce the concept of coordinate coding [9].

Definition: A coordinate coding is a pair (γ, C), where $C \subset \mathbf{R}^d$ is a set of anchor points, and γ is a map of $\mathbf{x} \in \mathbf{R}^d$ to $[\gamma_v(\mathbf{x})]_{v \in C} \in R^{|C|}$. It induces the following physical approximation of $\mathbf{x}$ in $\mathbf{R}^d : \gamma(\mathbf{x}) = \sum_{v \in C} \gamma_v(\mathbf{x}) v$.

In order to enhance the sparseness of NMF, Chen *et al.* incorporated the local coordinate constraint into NMF and pro-

posed NLCF [1]. The objective function of NLCF is defined as follows:

$$J = \sum_{i=1}^{n}(\|\mathbf{x}_i - \mathbf{U}\mathbf{v}_i\|^2 + \mu\sum_{c=1}^{k}|v_{ci}|\|\mathbf{u}_c - \mathbf{x}_i\|^2) \quad (2)$$

where $\mathbf{x}_i$ denotes the i-th column of $\mathbf{X}$, $\mathbf{v}_i$ is the i-th column of $\mathbf{V}$, $\mathbf{u}_c$ is the c-th column of $\mathbf{U}$, v_{ci} is the coordinate of $\mathbf{x}_i$ with respect to $\mathbf{u}_c$. The second term in (2) is the local coordinate factorization regularization. The columns of the basis matrix $\mathbf{U}$ can be regarded as a set of anchor points, and each data point can be represented by a linear combination of the anchor points. The coordinates of the data points with respect to the anchor points are represented by the columns of $\mathbf{V}$. The local coordinate factorization regularization forces each data point to be represented by only few anchor points so as to obtain sparse coding [1].

3. NON-NEGATIVE LOCAL COORDINATE FACTORIZATION WITH PAIRWISE CONSTRAINTS

3.1. The Objective Function

In order to improve the performance of NLCF, we incorporates pairwise constraints and graph Laplacian into NLCF, we expect that data points with pairwise must-link constraints should have the similar coordinates, while data points having pairwise cannot-link constraints should have distinct coordinates. Pairwise constraints can be user specified or generated among the randomly selected labeled points from the database. For simplicity, we exploit the ground-truth data labels to generated the pairwise constraints. When we have same labeled points in the database, we can obtain the specific pairwise constraints information among them. 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. Then we can construct a must-link pairwise constraint symmetric matrix $\mathbf{M} = [m_{pj}]$ $(p, j = 1, 2, \cdots, n)$ and a cannot-link pairwise constraint symmetric matrix $\mathbf{C} = [c_{pj}]$ $(p, j = 1, 2, \cdots, n)$ as follows:

$$m_{pj} = \begin{cases} 1 & \text{if } \mathbf{x}_p, \mathbf{x}_j (p \neq j) \text{ have the same class label} \\ 0 & \text{otherwise} \end{cases} \quad (3)$$

$$c_{pj} = \begin{cases} 1 & \text{if } \mathbf{x}_p, \mathbf{x}_j (p \neq j) \text{ have different class labels} \\ 0 & \text{otherwise} \end{cases} \quad (4)$$

With the pairwise constraints, our PCGNLCF algorithm aims to minimize the following objective function:

$$J = \sum_{i=1}^{m}\sum_{j=1}^{n}(x_{ij} - \sum_{c=1}^{k}u_{ic}v_{cj})^2 + \lambda\sum_{i=1}^{m}\sum_{j=1}^{n}\sum_{c=1}^{k}[|v_{cj}| \quad (5)$$

$$(u_{ic} - x_{ij})^2] + \alpha\frac{1}{2}\sum_{c=1}^{k}\sum_{q=1}^{n}\sum_{j=1}^{n}w_{jq}(v_{cj} - v_{cq})^2 +$$

$$\beta\sum_{j=1}^{n}(\sum_{p:m_{pj}=1}\sum_{c=1}^{k}\sum_{h=1,h\neq c}^{k}v_{cj}v_{hp} + \sum_{p:c_{pj}=1}\sum_{c=1}^{k}v_{cj}v_{cp})$$

$$w_{jq} = \begin{cases} \exp(-\frac{||\mathbf{x}_j - \mathbf{x}_q||^2}{\sigma^2}) & \text{if } \mathbf{x}_q \in N_p(\mathbf{x}_j) \\ & \text{or } \mathbf{x}_j \in N_p(\mathbf{x}_q) \text{ and } j \neq q \\ 0 & \text{otherwise} \end{cases} \quad (6)$$

The Eq. (5) can be rewritten in matrix form using an auxiliary matrix $\mathbf{A} \in \mathbf{R}^{k\times k}$, $\mathbf{A}$ is defined as:

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

$\mathbf{A}$ is a symmetric matrix, the diagonal elements of $\mathbf{A}$ are all 0, and other elements are all 1.

$$J = ||\mathbf{X} - \mathbf{U}\mathbf{V}||^2 + \lambda\mathrm{tr}[(\mathbf{S} - 2\mathbf{U}^T\mathbf{X} + \mathbf{Z})\mathbf{V}^T] \quad (7)$$
$$+\alpha\mathrm{tr}(\mathbf{V}\mathbf{L}\mathbf{V}^T) + \beta[\mathrm{tr}(\mathbf{A}\mathbf{V}\mathbf{M}\mathbf{V}^T) + \mathrm{tr}(\mathbf{V}\mathbf{C}\mathbf{V}^T)]$$

where $\mathbf{Z} = (\mathbf{z}, \cdots, \mathbf{z})^T$ is a $k \times n$ matrix, and $\mathbf{z} = diag(\mathbf{X}^T\mathbf{X}) \in \mathbf{R}^n$. $\mathbf{S} = (\mathbf{s}, \cdots, \mathbf{s})$ is a $k \times n$ matrix, and $\mathbf{s} = diag(\mathbf{U}^T\mathbf{U}) \in \mathbf{R}^k$ [1].

In Eq.(5), $N_p(\mathbf{x}_q)$ denotes the set of the p nearest neighbors of the data point $\mathbf{x}_q$. $u_{ic} \geq 0$ and $v_{cj} \geq 0$, $i = 1, 2, \cdots, m; q, p, j = 1, 2, \cdots, n; c = 1, 2, \cdots, k$. The first term in (5) corresponds to the cost function of NMF, it denotes the squared sum of the Euclidean distance between $\mathbf{X}$ and $\mathbf{U}\mathbf{V}$. The second term is the local coordinate constraint regularization, which forces each data point to be represented by only few anchor points and leads to sparse representations for the data. The third term is graph Laplacian regularization which is used to capture the local structure of the data. The fourth term is the cost function for violation of the pairwise constraints. We now analyze how the two components of pairwise constraints work with Eq.(5):

Suppose $\mathbf{x}_j$ has the largest coordinate v_{cj} with respect to the anchor point $\mathbf{u}_c$ in the j-th column of $\mathbf{V}$. If $\mathbf{x}_p$ has a must-link constraint with $\mathbf{x}_j$ $(m_{pj} = 1)$, then $\mathbf{x}_p$ should also have the largest coordinate v_{cp} with respect to the anchor point $\mathbf{u}_c$ in the p-th column of $\mathbf{V}$. In this case, the product of v_{cj} and v_{cp} is the biggest than any other product of v_{cj} and v_{hp} $(h = 1, ..., k; h \neq c)$ in the j-th column and the p-th column of $\mathbf{V}$. Therefore, v_{cp} should be

maximized in the p-th column of $\mathbf{V}$, this is imposed by minimizing $\sum_{j=1}^{n}(\sum_{p:m_{pj}=1}\sum_{c=1}^{k}\sum_{h=1,h\neq c}^{k}v_{cj}v_{hp})$, when it is minimized, $v_{hp}(h=1,...,k;h\neq c)$ will be as smaller as possible, while v_{cp} will be getting bigger as much as possible. Eventually, $\mathbf{x}_p$ will have the largest coordinate v_{cp} in the p-th column of $\mathbf{V}$. When $\mathbf{x}_j$ and $\mathbf{x}_p$ have a cannot-link constraint ($c_{pj}=1$), they should have distinct coordinates. That is, the j-th column and the p-th column of $\mathbf{V}$ are as orthogonal as possible. This can be imposed by minimizing $\sum_{j=1}^{n}(\sum_{p:c_{pj}=1}\sum_{c=1}^{k}v_{cj}v_{cp})$.

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

3.2. The Algorithm

The objective function *J* of PCGNLCF in Eq. (7) is not convex in both two matrices 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 based on the Lagrangian Multiplier method which can obtain a local optimum of *J*.

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

$$\mathcal{L}=J+\mathrm{tr}(\mathbf{\Phi}\mathbf{U}^T)+\mathrm{tr}(\mathbf{\Psi}\mathbf{V}^T) \tag{8}$$

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

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{V}}&=-2(1+\lambda)(\mathbf{U}^T\mathbf{X})+2(\mathbf{U}^T\mathbf{U}\mathbf{V})+\lambda(\mathbf{S}+\mathbf{Z}) \\ &+2\alpha(\mathbf{V}\mathbf{D})-2\alpha(\mathbf{V}\mathbf{W})+\beta(\mathbf{A}\mathbf{V}\mathbf{M}+\mathbf{V}\mathbf{C})+\mathbf{\Psi}=0\end{aligned} \tag{9}$$

$$\begin{aligned}\frac{\partial\mathcal{L}}{\partial\mathbf{U}}&=-2(1+\lambda)(\mathbf{X}\mathbf{V}^T)+2\lambda(\mathbf{U}\mathbf{H})+2(\mathbf{U}\mathbf{V}\mathbf{V}^T) \\ &+\mathbf{\Phi}=0\end{aligned} \tag{10}$$

where $\mathbf{H}\in\mathbf{R}^{k\times k}$. $\mathbf{H}$ is a diagonal matrix with its entries defined as $h_{cc}=\sum_{j=1}^{n}v_{cj}$. Using the KKT conditions $\psi_{cj}v_{cj}=0$ and $\phi_{ic}u_{ic}=0$, we get the following equations for v_{cj} and u_{ic}:

$$v_{cj}\longleftarrow v_{cj}\frac{2(1+\lambda)(\mathbf{U}^T\mathbf{X})_{cj}+2\alpha(\mathbf{V}\mathbf{W})_{cj}}{f(\mathbf{U},\mathbf{V})} \tag{11}$$

$$\begin{aligned}f(\mathbf{U},\mathbf{V})&=2(\mathbf{U}^T\mathbf{U}\mathbf{V})_{cj}+\lambda(\mathbf{S}+\mathbf{Z})_{cj}+2\alpha(\mathbf{V}\mathbf{D})_{cj} \\ &+\beta(\mathbf{A}\mathbf{V}\mathbf{M}+\mathbf{V}\mathbf{C})_{cj}\end{aligned} \tag{12}$$

$$u_{ic}\longleftarrow u_{ic}\frac{(1+\lambda)(\mathbf{X}\mathbf{V}^T)_{ic}}{(\mathbf{U}\mathbf{V}\mathbf{V}^T)_{ic}+\lambda(\mathbf{U}\mathbf{H})_{ic}} \tag{13}$$

The convergence proof and computational complexity of PCGNLCF have been provided in the supplementary material.

3.3. Computational Complexity Analysis

The objective function of the PCGNLCF is minimized by iteratively updating matrices $\mathbf{U}$ and $\mathbf{V}$. In this subsection, we will discuss the extra computational cost of our PCGNLCF algorithm.

The big O analysis is usually used to express the complexity of the algorithm [8]. However, it may be not precise enough to differentiate the complexity of PCGNLCF [10]. Thus, we count the arithmetic operations for PCGNLCF algorithm [10, 8, 11]. Three arithmetic operations addition, multiplication and division are involved in the updating computation. All these operations are performed on floating-point numbers [8]. 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 PCGNLCF. It is important to note that $\mathbf{M}$ and $\mathbf{C}$ 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 $\mathbf{AVM}$ and C_NK flam to compute $\mathbf{VC}$. Moreover, $\mathbf{W}$ is also a sparse matrix, we only need NpK flam to compute $\mathbf{VW}$ [10]. $\mathbf{S}$ and $\mathbf{Z}$ will cost MK, MN flam respectively [11], $\mathbf{H}$ will cost NK fladd (a floating point addition).

So PCGNLCF needs $(2MNK+3MK+8NK+MN+2MK^2+2NK^2+NpK+M_NK+C_NK)$ fladd, $(2MNK+3MK+2NK+MN+2MK^2+2NK^2+NpK+M_NK+C_NK)$ flmlt (a floating point multiplication) and $(MK+NK)$ fldiv (a floating point division) in each iteration. Besides the multiplicative updating, PCGNLCF needs $O(L^2)$ to construct the constraint matrices $\mathbf{M}$ and $\mathbf{C}$, and PCGNLCF also needs $O(N^2M)$ to construct the p-nearest neighbor graph [10].

If the multiplicative updates stop after t iterations, the overall computational complexity for PCGNLCF will be $O(tMNK+L^2+N^2M)$.

4. EXPERIMENTAL RESULTS

In this section, we use image clustering tasks to evaluate our PCGNLCF algorithm. Two metrics are used to evaluate the clustering performance of each experiment. The algorithms are evaluated by comparing the cluster label of each sample point with its label provided by the dataset. The first metric is

Table 1. Parameters Used in Complexity Analysis

Parameters	Description
N	number of data points
M	number of features
K	number of factors
L	number of labeled data points
M_N	number of pairwise must-link constraints
C_N	number of pairwise cannot-link constraints

the accuracy (*AC*), which can be used to measure the percentage of correct labels obtained by an algorithm. *AC* is defined as follows [12], its details can refer to [10].

$$AC = \frac{\sum_{i=1}^{n} \delta(\gamma_i, map(l_i))}{n} \tag{14}$$

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 second metric is the normalized mutual information (NMI). In clustering problems, normalized mutual information can measure how similar two clusters are. NMI is defined as follows, its details can refer to [10].

$$NMI(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))} \tag{15}$$

Where $MI(C, C')$ denotes the mutual information and $H(C)$ is the entropy.

The sparseness measure is based on the relationship between the L_1 norm and the L_2 norm [13, 11], it is defined as follows:

$$SP(\mathbf{v}) = \frac{\sqrt{N} - (\sum |v_i| / \sqrt{\sum v_i^2})}{\sqrt{N} - 1} \tag{16}$$

where $\mathbf{v}$ is a column vector of $\mathbf{V}$, N is the dimensionality of $\mathbf{v}$. If all entry of $\mathbf{v}$ are equal, $SP(\mathbf{v})$ takes a value of zero. If $\mathbf{v}$ contains only a single non-zero entry, $SP(\mathbf{v})$ evaluates to unity. In our experiments, we report the average sparseness on ten independent experiments on each database. On each independent experiment, we compute the average sparseness over all the new representations with each column vectors of $\mathbf{V}$ [11].

4.1. Performance Evaluations and Comparisons

To evaluate how the clustering performance can be improved by our method, we compare our algorithm with other state-of-the-art algorithms, such as NMF [14], NLCF [1], NMFSC [13], GNMF [10], CNMF [8], SSNMF [7], GNLCF (PCGNLCF with setting β = 0) [1], PCNLCF (PCGNLCF with setting $\alpha = 0$).

Experiments are performed on four image databases, UMIST [1] database consists of 575 images of 20 individuals, the size of each cropped image is 40×40 pixels, with 256 gray levels per pixel, thus, each image is represented by a 1600-dimensional vector. Georgia Tech Face (GTF) [2] database obtains 750 images of 50 distinct subjects, each image is represented by a 2500-dimensional vector. COIL100 [3], USPS [4] handwritten digit database contains 10 objects. We select a popular subset containing 9298 16×16 handwritten digit images in total.

We conduct ten independent experiments on each dataset, and the average clustering results and standard deviations are reported in Table 2. In each experiment, we randomly select 25 subjects for clustering on GTF and COIL100. On UMIST, 10 subjects are randomly selected for clustering. On USPS, we randomly select 6 subjects for clustering in each experiment. In our experiments, four images are randomly selected from each cluster with labels on UMIST. On GTF, we randomly select five images from each category to provide the label information. For USPS and COIL100, we randomly pick up 10% images from each cluster as the available label information. For PCGNLCF, the pairwise constraints are generated among all the labeled images on each database. We set k to be the number of clusters [1]. In the clustering process, for NMF, NMFSC, GNMF and CNMF, in order to achieve the best performance, fast K-means algorithm [15] is further applied to the new data representation $\mathbf{V}$ for clustering [10, 8]. For NLCF, GNLCF, PCNLCF and PCGNLCF, we use $\mathbf{V}$ to determine the cluster label of each data point [1]. That is, we examine each column of $\mathbf{V}$, and assign data point $\mathbf{x}_j$ to cluster c if $c = \arg\max_c v_{cj}$.

For each algorithm, in order to achieve its best results, the parameters are appropriately selected. For PCGNLCF, λ searches the grid $\{0.1, 0.2, 0.5, 1\}$, α is set by searching the grid $\{0.1, 0.5, 1, 5, 10\}$, β searches the grid $\{1, 5, 10, 15\}$. On UMIST and COIL100, we fix $\lambda = 0.2, \alpha$ = 10, β = 5, on GTF we set $\lambda = 0.2, \alpha$ = 0.1, β = 5, for USPS, we let $\lambda = 0.2, \alpha$ = 1, β = 5. The number of the nearest neighbors p searches the grid $\{3, 4, 5, 6, 7, 8, 9, 10\}$. More results and parameters settings of other algorithms can be found in the supplementary material.

When $\beta = 0$, PCGNLCF degenerates to GNLCF, when $\alpha = 0$, PCGNLCF degenerates to PCNLCF. PCGNLCF degenerates to NLCF when $\alpha = 0$ and $\beta = 0$. Table 2 to Table 3 show the detailed clustering accuracy, normalized mutual information and standard deviation on all the datasets. The average sparseness of the coefficients matrix obtained by the algorithms has been shown in Table 4. It can be seen that our proposed PCGNLCF algorithm consistently outperforms NMF, NLCF, NMFSC, GNMF, CNMF, SSNMF, GNLCF and

[1] http://www.sheffield.ac.uk/eee/research/iel/research/face

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

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

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

Table 2. Clustering Accuracy Comparison on the four Images Databases

Methods	Accuracy(%)			
	UMIST	GTF	COIL100	USPS
NMF	47.1±3.8	49.1±2.2	59.8±5.0	71.5±5.8
NLCF	46.9±4.4	49.9±4.2	55.6±5.9	67.0±5.1
NMFSC	49.2±5.1	49.5±2.2	59.6±5.7	71.1±6.6
GNMF	69.7±4.9	47.0±3.0	72.3±5.7	77.1±7.3
CNMF	51.9±3.4	56.2±3.2	62.6±4.5	76.3±6.6
SSNMF	60.1±4.6	75.4±2.7	68.2±5.7	83.1±2.1
GNLCF	55.8±4.6	53.7±4.0	62.9±4.3	77.8±9.3
PCNLCF	54.3±6.9	75.5±1.9	58.4±4.7	79.1±6.0
PCGNLCF	**90.2±4.5**	**81.6±2.7**	**91.4±2.6**	**96.2±1.1**

Table 3. Clustering normalized mutual information comparison on the four images databases

Methods	Normalized Mutual Information(%)			
	UMIST	GTF	COIL100	USPS
NMF	56.6±5.2	60.2±2.0	72.4±4.9	58.5±3.9
NLCF	58.6±4.8	63.6±3.5	71.4±5.1	54.8±4.6
NMFSC	58.0±6.3	62.5±1.8	73.2±5.1	58.1±5.1
GNMF	78.2±4.7	59.6±3.2	86.1±3.9	76.2±3.5
CNMF	60.6±2.5	67.5±2.3	74.3±4.3	63.0±4.1
SSNMF	62.1±5.7	76.2±2.6	73.7±4.8	65.2±3.3
GNLCF	67.5±3.5	65.2±2.9	76.9±4.3	72.0±6.9
PCNLCF	59.8±6.4	76.8±1.8	68.8±3.8	62.5±4.7
PCGNLCF	**89.2±4.1**	**81.6±2.5**	**93.0±2.3**	**88.7±2.6**

PCNLCF on each database. On UMIST database, GNMF obtains the second best performance, PCGNLCF achieves 20.5% improvement in accuracy and 11% improvement in normalized mutual information over GNMF on average. On USPS database, SSNMF obtains the second best result for accuracy measurement on average, GNMF gets the second best performance in normalized mutual information measurement on average. PCGNLCF achieves 13.1% improvement than SSNMF in accuracy and 12.5% improvement over GNMF in normalized mutual information on average. GNLCF only considers the local structure of the data. PCNLCF only considers the pairwise constraints. From Table 2 to Table 3 we can see that both the two algorithms can not achieve the best performance. With consideration of graph Laplacian and pairwise constraints, PCGNLCF achieves the best performance and outperforms others algorithms significantly.

Table 4 shows the average sparseness of the coefficients matrix obtained by NMF, NLCF CNMF, SSNMF, GNLCF, PCNLCF and PCGNLCF. We do not list the sparseness obtained by NMFSC and GNMF, because if NMFSC and GNMF can get a better performance, the sparseness obtained by NMFSC and GNMF will be lower than NMF in some cases according to our experiments. When the sparseness obtained by NMFSC and GNMF increases, the performances of NMFSC and GNMF will decrease. In our paper, we make a balanced choice between the performances and sparseness. From Table 4, we can see that NLCF, GNLCF, PCNLCF and PCGNLCF can indeed achieve sparser representations for the data. Mainly because these algorithms have incorporated the local coordinate constraint into NMF.

Table 4. Average sparseness of coefficients matrix on each database

Methods	Sparseness(%)			
	UMIST	GTF	COIL100	USPS
NMF	45.2±1.7	36.4±0.7	48.7±2.2	32.9±1.2
NLCF	**92.7±0.8**	90.3±0.6	**95.6±0.7**	73.3±2.1
CNMF	42.1±2.0	35.1±0.6	46.6±2.2	31.6±1.8
SSNMF	73.9±1.4	86.7±0.6	79.5±1.6	65.1±1.4
GNLCF	77.5±2.2	65.6±2.8	90.5±1.8	73.2±2.5
PCNLCF	87.4±1.3	**91.7±0.5**	87.3±1.0	72.3±1.3
PCGNLCF	79.7±3.2	87.1±0.9	81.4±3.1	**82.1±2.6**

Moreover, it is interesting to see that even if NLCF can obtain sparser representations for data in most instances, but it does not necessarily have the best performance. This is mainly because that NLCF can not guarantee that data points sharing the same class label will be as close to the same anchor point as much as possible, while data points possessing different class labels will be as close to different anchor points as much as possible.

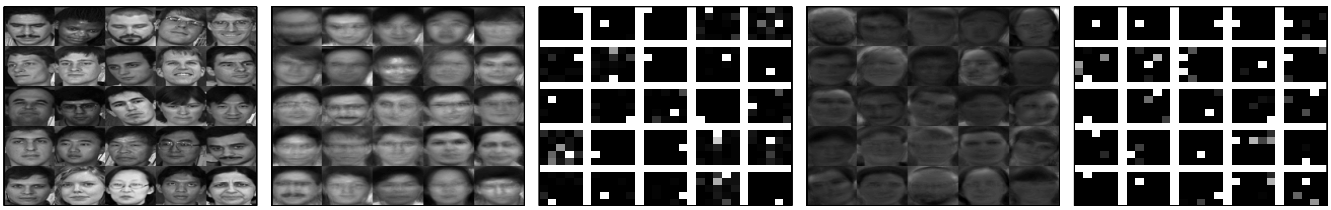
The sample images from GTF has been shown in Fig. 1 (a). Fig. 1 (b) to Fig. 1 (e) have shown the bases vectors and image encodings obtained by PCGNLCF and NLCF. We can see that PCGNLCF learns better bases vectors. The bases vectors learned by PCGNLCF are more clear and bright, and they look more like the original face images. The image encoding of each image obtained by PCGNLCF has almost only one non-zero entry. The non-zero entry is the coordinate coefficient with respect to the basis image, which is closest to the face image [1].

5. CONCLUSION

In this paper, PCGNLCF is proposed. PCGNLCF considers pairwise constraints of the data and graph Laplacian. PCGNLCF can enhance the discriminative powerful of the new representations of the data and achieve better performance.

6. ACKNOWLEDGMENTS

This work is supported by NSFC (No.61272247, 60873133), 863 (No.2008AA02Z310, SS2015AA020501) in China, the Science and Technology Commission of Shanghai Municipality (Grant No.13511500200) and the Innovation Ability Special Fund of Shanghai Jiao Tong University under Grant Z030026.



(a) (b) (c) (d) (e)

Fig. 1. (a) is the original images on Georgia Tech Face, which contains 25 images from different 25 subjects. (b) and (d) are the bases vectors learned by PCGNLCF and NLCF respectively. (c) and (e) are the image encodings (the obtained new representations of the data) of PCGNLCF and NLCF respectively. Non-zero values are shown with white pixels.

7. REFERENCES

[1] Yan Chen, Jiemi Zhang, Deng Cai, Wei Liu, and Xiaofei He, "Nonnegative local coordinate factorization for image representation," *Image Processing, IEEE Transactions on*, vol. 22, no. 3, pp. 969–979, 2013.

[2] D.D. Lee and H.S. Seung, "Algorithms for non-negative matrix factorization," *Advances in neural information processing systems*, vol. 13, 2001.

[3] Matthias Heiler and Christoph Schnörr, "Learning sparse representations by non-negative matrix factorization and sequential cone programming," *The Journal of Machine Learning Research*, vol. 7, pp. 1385–1407, 2006.

[4] O. Chapelle, B. Schölkopf, A. Zien, et al., *Semi-supervised learning*, vol. 2, MIT press Cambridge, MA:, 2006.

[5] N. Grira, M. Crucianu, and N. Boujemaa, "Semi-supervised fuzzy clustering with pairwise-constrained competitive agglomeration," in *Fuzzy Systems, 2005. FUZZ'05. The 14th IEEE International Conference on*. IEEE, 2005, pp. 867–872.

[6] R. He, W.S. Zheng, B.G. Hu, and X.W. Kong, "Nonnegative sparse coding for discriminative semi-supervised learning," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 2849–2856.

[7] H. Lee, J. Yoo, and S. Choi, "Semi-supervised nonnegative matrix factorization," *Signal Processing Letters, IEEE*, vol. 17, no. 1, pp. 4–7, 2010.

[8] Haifeng Liu, Zhaohui Wu, Xuelong Li, Deng Cai, and Thomas S Huang, "Constrained nonnegative matrix factorization for image representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 34, no. 7, pp. 1299–1311, 2012.

[9] Kai Yu, Tong Zhang, and Yihong Gong, "Nonlinear learning using local coordinate coding," *Advances in Neural Information Processing Systems*, vol. 22, pp. 2223–2231, 2009.

[10] D. Cai, X. He, J. Han, and T.S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 33, no. 8, pp. 1548–1560, 2011.

[11] Yan Chen, Hujun Bao, and Xiaofei He, "Non-negative local coordinate factorization for image representation," in *Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on*. IEEE, 2011, pp. 569–574.

[12] L. Lovász and M.D. Plummer, *Matching theory*, Number 121. Elsevier Science Ltd, 1986.

[13] P.O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *The Journal of Machine Learning Research*, vol. 5, pp. 1457–1469, 2004.

[14] W. Xu, X. Liu, and Y. Gong, "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, 2003, pp. 267–273.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Computer Vision and Pattern Recognition, 2007. CVPR'07. IEEE Conference on*. Ieee, 2007, pp. 1–8.