Nonnegative Local Coordinate Factorization for Image Representation

Yan Chen, Jiemi Zhang, Deng Cai, *Member, IEEE*, Wei Liu, and Xiaofei He, *Senior Member, IEEE*

***Abstract*— Recently, nonnegative matrix factorization (NMF) has become increasingly popular for feature extraction in computer vision and pattern recognition. NMF seeks two nonnegative matrices whose product can best approximate the original matrix. The nonnegativity constraints lead to sparse parts-based representations that can be more robust than nonsparse global features. To obtain more accurate control over the sparseness, in this paper, we propose a novel method called nonnegative local coordinate factorization (NLCF) for feature extraction. NLCF adds a local coordinate constraint into the standard NMF objective function. Specifically, we require that the learned basis vectors be as close to the original data points as possible. In this way, each data point can be represented by a linear combination of only a few nearby basis vectors, which naturally leads to sparse representation. Extensive experimental results suggest that the proposed approach provides a better representation and achieves higher accuracy in image clustering.**

***Index Terms*— Local coordinate coding, nonnegative matrix factorization, sparse learning.**

I. Introduction

DATA REPRESENTATION has multiple meanings and goals, arising from many applications, such as computer vision and pattern recognition. In these fields, the input data matrix is often of very high dimension, which may make *learning from example* infeasible [1]. One hopes to reduce the dimension by using feature extraction techniques, such as Principal Component Analysis (PCA), Singular Value Decomposition (SVD), Non-negative Matrix Factorization (NMF, [2]) and other methods [3]–[5].

PCA is one of the most popular dimensionality reduction methods. Its operation can be thought of as revealing the internal structure of the data in a way which best explains the variance in the data. The basis of PCA are orthogonal and have a statistical interpretation as the directions of largest variance. Recently, matrix factorization techniques have become increasing popular for feature extraction. One of the most frequently used matrix factorization techniques is SVD, which provides a low-rank approximation to the original matrix. This approximation is optimal in the sense of reconstruction error and thus optimal for data representation when Euclidean structure is concerned.

Unlike PCA and SVD, NMF seeks for two *non-negative* matrices whose product can best approximate the original matrix. Previous studies have shown there is psychological and physiological evidence for parts-based representation in human brain [6]–[8]. The NMF codes naturally favor sparse, parts-based representations which in the context of classification and regression can be more robust than non-sparse, global representations [9]. Due to the non-negativity constraints of NMF, it models each data point as additive, not subtractive, combination of the underlying clusters. However, NMF does not always result in sparse representation [10]. Hoyer extended NMF to include the option to control sparseness explicitly by adding a $L1$ norm minimization on the factor matrices, which allows us to discover sparse representations better than those given by standard NMF [11]. It would be important to note that, however, Hoyer's approach does not directly ensure the sparseness of the new representation of a data point. Instead, it ensures the sparseness of a new feature corresponding to a basis vector. Thus, although in average the new representations of the data points can be very sparse, theoretically it is possible that the new representations for some points are highly sparse while for the others the new representations are highly dense.

In this paper we propose a novel matrix factorization algorithm, called Non-negative Local Coordinate Factorization (NLCF), which adds a local coordinate constraint to ensure the sparseness of the obtained representations. Our algorithm is motivated by many recent progresses on sparse coding, and particularly, Local Coordinate Coding proposed by Yu et al. [12], [13]. Specifically, we require that the learned basis vectors be as close to the data points as possible. In this way, each data point can be represented by a linear combination of only few nearby basis vectors and, thus, the sparseness of the obtained representations can be guaranteed. An optimization scheme has been developed to solve the objective function based on iterative updates of the two factor matrices. It is important to note that NLCF is unsupervised, which is fundamentally different form [14] which is supervised.

Manuscript received March 6, 2012; revised July 19, 2012; accepted September 29, 2012. Date of publication October 12, 2012; date of current version January 24, 2013. This work was supported in part by the National Basic Research Program of China (973 Program) under Grant 2013CB336500, and the National Natural Science Foundation of China under Grant 61222207, Grant 61125203, and Grant 91120302. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Mark Liao. *(Corresponding author: D. Cai.)*

Y. Chen, J. Zhang, D. Cai, and X. He are with the State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, Zhejiang 310058, China (e-mail: yanchen036@gmail.com; jmzhang10@gmail.com; dengcai@cad.zju.edu.cn; xiaofeihe@cad.zju.edu.cn).

W. Liu is with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY 10598 USA (e-mail: wliu@ee.columbia.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TIP.2012.2224357

The rest of the paper is organized as follows: Section 2 gives a brief review of NMF. In Section 3, we introduce our NLCF algorithm, as well as the optimization scheme, convergence study and computational complexity analysis. Extensive experimental results are presented in Section 4. Finally, we conclude in Section 5.

II. Brief Review of NMF

NMF tries to decompose a non-negative $M \times N$ matrix $\mathbf{X}$ into two non-negative factor matrices $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_K) \in \mathbb{R}^{M\times K}$ and $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_N) \in \mathbb{R}^{K\times N}$. There are different criteria to measure the quality of the decomposition. Lee et al. proposed two objective functions in [15]: the Euclidean distance between $\mathbf{X}$ and $\mathbf{UV}$ [16] and the KL divergence [7]. The Euclidean distance based objective function is expressed as:

$$\mathcal{O} = \|\mathbf{X} - \mathbf{UV}\|_F^2 \tag{1}$$

where $||\cdot||_F$ denotes the matrix *Frobenius norm*.

Because the objective function $\mathcal{O}$ is not convex in both $\mathbf{U}$ and $\mathbf{V}$, it is infeasible to find a global minimum of $\mathcal{O}$ [17]. The following iterative update rules provided by Lee et al. [15] can obtain a local minimum of $\mathcal{O}$:

$$u_{jk}^{t+1} = u_{jk}^t \frac{(\mathbf{XV^T})_{jk}}{(\mathbf{UVV}^T)_{jk}}$$
$$v_{ki}^{t+1} = v_{ki}^t \frac{(\mathbf{U}^T\mathbf{X})_{ki}}{(\mathbf{U}^T\mathbf{UV})_{ki}}.$$

By NMF, each data point $\mathbf{x}_i$ is approximated by a linear combination of the columns of $\mathbf{U}$, weighted by the elements of the *i-th* column of $\mathbf{V}$. Please see [18]–[24] for various NMF extensions. The NMF has been successfully used in many multimedia applications [25], [26] and the close technique probabilistic latent semantic analysis [27] has also been widely discussed [28].

III. Nonnegative Local Coordinate Factorization

In this section, we introduce our NLCF algorithm for obtaining sparse representation.

A. Objective Function

We first introduce the concept of coordinate coding [12].

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

By this definition, the columns of the basis matrix $\mathbf{U}$ can be considered as a set of anchor points, and each data point in the original space can be approximated by a linear combination of the anchor points. The columns of $\mathbf{V}$ contains the coordinates of the data points with respect to the anchor points.

In order to obtain sparse codings, each data point should be represented as a linear combination of only few nearby anchor points. In other words, each data point should be sufficiently close to only few anchor points. This can be achieved by introducing the local coordinate constraint [12]:

$$\mathcal{Q} = \sum_{k=1}^{K} |v_{ki}| \, \|\mathbf{u}_k - \mathbf{x}_i\|^2 . \tag{2}$$

The above constraint incurs a heavy penalty if $\mathbf{x}_i$ is far away from the anchor point $\mathbf{u}_k$ while its new coordinate v_{ki} with respect to $\mathbf{u}_k$ is large. Therefore, minimizing it is an attempt to ensure that if $\mathbf{x}_i$ is sufficiently close to the anchor point $\mathbf{u}_k$ then its new coordinate with respect to $\mathbf{u}_k$ tends to be one.

By incorporating the local coordinate constraint Q into the standard NMF objective function, we get the following minimization problem:

$$\begin{aligned} \mathcal{O} = &\sum_{i=1}^{N} \Big(\|\mathbf{x}_i - \mathbf{U}\mathbf{v}_i\|^2 + \mu \sum_{k=1}^{K} |v_{ki}| \, \|\mathbf{u}_k - \mathbf{x}_i\|^2 \Big) \\ \text{s.t.} \quad &\mathbf{U} = [\mathbf{u}_1, \ldots, \mathbf{u}_K] \in \mathbb{R}^{M\times K} > 0 \\ &\mathbf{V} = [\mathbf{v}_1, \ldots, \mathbf{v}_N] \in \mathbb{R}^{K\times N} > 0 \end{aligned} \tag{3}$$

where $\mu \geq 0$ is a regularization parameter.

B. Update Rules

Following some simple algebraic steps, we can rewrite the objective function as follows:

$$\begin{aligned} \mathcal{O} &= \|\mathbf{X} - \mathbf{UV}\|^2 + \sum_{i=1}^{N} \mu \sum_{k=1}^{K} |v_{ki}| \, \|\mathbf{u}_k - \mathbf{x}_i\|^2 \\ &= \|\mathbf{X} - \mathbf{UV}\|^2 + \mu \sum_{i=1}^{N} \left\| (\mathbf{x}_i \mathbf{1}^T - \mathbf{U}) \Lambda_i^{1/2} \right\|^2 \end{aligned}$$

where $\Lambda_i = diag(|\mathbf{v}_i|) \in \mathbb{R}^{K\times K}$.

Noticing that $\|A\|^2 = \mathrm{Tr}(AA^T)$, we have

$$\begin{aligned} \mathcal{O} = \mathrm{Tr}\Big(&(\mathbf{X} - \mathbf{UV})(\mathbf{X} - \mathbf{UV})^T \\ &+ \mu \sum_{i=1}^{N} (\mathbf{x}_i \mathbf{1}^T - \mathbf{U}) \Lambda_i (\mathbf{x}_i \mathbf{1}^T - \mathbf{U})^T \Big) \\ = \mathrm{Tr}\Big(&\mathbf{XX}^T + \mathbf{UVV}^T\mathbf{U}^T - 2\mathbf{XV}^T\mathbf{U}^T \\ &+ \mu \sum_{i=1}^{N} (\mathbf{x}_i \mathbf{1}^T \Lambda_i \mathbf{1}\mathbf{x}_i^T - 2\mathbf{x}_i \mathbf{1}^T \Lambda_i \mathbf{U}^T + \mathbf{U}\Lambda_i\mathbf{U}^T) \Big). \end{aligned} \tag{4}$$

Let ψ_{jk} and ϕ_{ki} be the Langrange multiplier for constraints $u_{jk} \geq 0$ and $v_{ki} \geq 0$, respectively. We define matrix $\Psi = [\psi_{jk}]$ and $\Phi = [\phi_{ki}]$, then the Langrange $\mathcal{L}$ is

$$\begin{aligned} \mathcal{L} = \mathrm{Tr}\Big(&\mathbf{XX}^T + \mathbf{UVV}^T\mathbf{U}^T - 2\mathbf{XV}^T\mathbf{U}^T \\ &+ \mu \sum_{i=1}^{N} (\mathbf{x}_i \mathbf{1}^T \Lambda_i \mathbf{1}\mathbf{x}_i^T - 2\mathbf{x}_i \mathbf{1}^T \Lambda_i \mathbf{U}^T + \mathbf{U}\Lambda_i\mathbf{U}^T) \Big) \\ &+ \mathrm{Tr}(\Psi\mathbf{U}^T) + \mathrm{Tr}(\Phi\mathbf{V}^T) \end{aligned}$$

The partial derivatives of $\mathcal{L}$ with respect to $\mathbf{U}$ and $\mathbf{V}$ are:

$$\begin{aligned} \frac{\partial \mathcal{L}}{\partial \mathbf{U}} = &2\mathbf{UVV}^T - 2\mathbf{XV}^T \\ &+ \mu \sum_{i=1}^{N} (-2\mathbf{x}_i \mathbf{1}^T \Lambda_i + 2\mathbf{U}\Lambda_i) + \Psi \end{aligned} \tag{5}$$

$$\frac{\partial \mathcal{L}}{\partial \mathbf{V}} = 2\mathbf{U}^T\mathbf{UV} - 2\mathbf{U}^T\mathbf{X} + \mu(\mathbf{C} - 2\mathbf{U}^T\mathbf{X} + \mathbf{D}) + \Phi. \tag{6}$$

Define column vector $\mathbf{c} = diag(\mathbf{X}^T\mathbf{X}) \in \mathbb{R}^N$. Let $\mathbf{C} = (\mathbf{c}, \ldots, \mathbf{c})^T$ be a $K \times N$ matrix whose rows are $\mathbf{c}^T$. Define column vector $\mathbf{d} = diag(\mathbf{U}^T\mathbf{U}) \in \mathbb{R}^K$. Let $\mathbf{D} = (\mathbf{d}, \ldots, \mathbf{d})$ be a $K \times N$ matrix whose columns are $\mathbf{d}$.

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

$$(\mathbf{U}\mathbf{V}\mathbf{V}^T)_{jk}u_{jk} - (\mathbf{X}\mathbf{V}^T)_{jk}u_{jk} + \mu(\sum_{i=1}^{N}\mathbf{U}\Lambda_i)_{jk}u_{jk} - \mu(\sum_{i=1}^{N}\mathbf{x}_i\mathbf{1}^T\Lambda_i)_{jk}u_{jk} = 0$$

$$2(\mathbf{U}^T\mathbf{U}\mathbf{V})_{ki}v_{ki} - 2(\mathbf{U}^T\mathbf{X})_{ki}v_{ki} + \mu(\mathbf{C} - 2\mathbf{U}^T\mathbf{X} + \mathbf{D})_{ki}v_{ki} = 0$$

The above equations lead to the following update rules:

$$u_{jk} \leftarrow u_{jk}\frac{(\mathbf{X}\mathbf{V}^T + \mu\sum_{i=1}^{N}\mathbf{x}_i\mathbf{1}^T\Lambda_i)_{jk}}{(\mathbf{U}\mathbf{V}\mathbf{V}^T + \mu\sum_{i=1}^{N}\mathbf{U}\Lambda_i)_{jk}} \tag{7}$$

$$v_{ki} \leftarrow v_{ki}\frac{2(\mu+1)(\mathbf{U}^T\mathbf{X})_{ki}}{(2\mathbf{U}^T\mathbf{U}\mathbf{V} + \mu\mathbf{C} + \mu\mathbf{D})_{ki}} \tag{8}$$

we will guarantee that the update rules of $\mathbf{U}$ and $\mathbf{V}$ in Eq. (7) and Eq. (8) converge and the final solution will be a local optimum. Please see the Appendix for a detailed proof.

C. Connection With Gradient Method

Here we will reveal the connection between Gradient Descent method [29] and our multiplicative updating rules in Eq. (7) and Eq. (8). Let $\eta_{jk} = -u_{jk}/2(\mathbf{U}\mathbf{V}\mathbf{V}^T + \mu\sum_{i=1}^{N}\mathbf{U}\Lambda_i)_{jk}$, we have

$$\begin{aligned}
&u_{jk} + \eta_{jk}\frac{\partial\mathcal{O}}{\partial u_{jk}} \\
&= u_{jk} - \frac{u_{jk}}{2(\mathbf{U}\mathbf{V}\mathbf{V}^T + \mu\sum_{i=1}^{N}\mathbf{U}\Lambda_i)_{jk}}\frac{\partial\mathcal{O}}{\partial u_{jk}} \\
&= u_{jk} - \frac{u_{jk}}{2(\mathbf{U}\mathbf{V}\mathbf{V}^T + \mu\sum_{i=1}^{N}\mathbf{U}\Lambda_i)_{jk}} \\
&\quad\times\Big((2\mathbf{U}\mathbf{V}\mathbf{V}^T - 2\mathbf{X}\mathbf{V}^T \\
&\qquad + \mu\sum_{i=1}^{N}(-2\mathbf{x}_i\mathbf{1}^T\Lambda_i + 2\mathbf{U}\Lambda_i))_{jk}\Big) \\
&= u_{jk}\frac{(\mathbf{X}\mathbf{V}^T + \mu\sum_{i=1}^{N}\mathbf{x}_i\mathbf{1}^T\Lambda_i)_{jk}}{(\mathbf{U}\mathbf{V}\mathbf{V}^T + \mu\sum_{i=1}^{N}\mathbf{U}\Lambda_i)_{jk}}
\end{aligned}$$

similarly, let $\delta_{ki} = -v_{ki}/(2\mathbf{U}^T\mathbf{U}\mathbf{V} + \mu\mathbf{C} + \mu\mathbf{D})_{ki}$, we have

$$\begin{aligned}
&v_{ki} + \delta_{ki}\frac{\partial\mathcal{O}}{\partial v_{ki}} \\
&= v_{ki} - \frac{v_{ki}}{(2\mathbf{U}^T\mathbf{U}\mathbf{V} + \mu\mathbf{C} + \mu\mathbf{D})_{ki}}\frac{\partial\mathcal{O}}{\partial v_{ki}} \\
&= v_{ki} - \frac{v_{ki}}{(2\mathbf{U}^T\mathbf{U}\mathbf{V} + \mu\mathbf{C} + \mu\mathbf{D})_{ki}} \\
&\quad\times\Big((2\mathbf{U}^T\mathbf{U}\mathbf{V} - 2\mathbf{U}^T\mathbf{X} + \mu(\mathbf{C} - 2\mathbf{U}^T\mathbf{X} + \mathbf{D}))_{ki}\Big) \\
&= v_{ki}\frac{2(\mu+1)(\mathbf{U}^T\mathbf{X})_{ki}}{(2\mathbf{U}^T\mathbf{U}\mathbf{V} + \mu\mathbf{C} + \mu\mathbf{D})_{ki}}.
\end{aligned}$$

TABLE I
ABBREVIATIONS FOR REPORTING OPERATION COUNTS

Abbreviations	Description
fladd	a floating-point addition
flmlt	a floating-point multiplication
fldiv	a floating-point division
flam	an addition and multiplication

TABLE II
COMPUTATIONAL OPERATION COUNTS FOR EACH MATRICES' MULTIPLICATION

	Fladd	Dlmlt
$\mathbf{X}\mathbf{V}^T$	MNK	MNK
$\mathbf{U}\mathbf{V}\mathbf{V}^T$	$(M+N)K^2$	$(M+N)K^2$
$\mathbf{U}\mathbf{H}$	$MK^2 + NK$	MK^2
$\mathbf{U}^T\mathbf{X}$	MNK	MNK
$\mathbf{U}^T\mathbf{U}\mathbf{V}$	$(M+N)K^2$	$(M+N)K^2$
$\mathbf{C}$	MN	MN
$\mathbf{D}$	MK	MK

N: the number of sample points
M: the number of features
K: the number of factors

Now it is clear that the multiplicative updating rules in Eq. (7) and Eq. (8) are special cases of gradient descent with automatically step parameter selection. The advantage of multiplicative updating rules is the guarantee of the non-negativity of $\mathbf{U}$ and $\mathbf{V}$.

D. Computational Complexity Analysis

In this section, a computational complexity analysis of our proposed algorithm comparing to NMF is presented.

The common way to express the complexity of one algorithm is using the big O notation [30]. However, it is not an appropriate way to analyze the complexity of an algorithm that contains many matrix computations such as NLCF and NMF. Instead, we count the arithmetic operations for each algorithm. The four operation abbreviations used in this paper are summarized in Table I. Please see [31] for more details about these operation abbreviations.

Based on the updating rules, we count the arithmetic operations of each iteration in NMF and summarize the result in Table III.

For NLCF, note that

$$\sum_{i=1}^{N}\mathbf{x}_i\mathbf{1}^T\Lambda_i = \mathbf{X}\mathbf{V}^T$$

$$\sum_{i=1}^{N}\mathbf{U}\Lambda_i = \mathbf{U}\mathbf{H}$$

where $\mathbf{H}$ is diagonal matrix whose entries are row sums of $\mathbf{V}$. So we can rewrite Eq. (7) as follows:

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

TABLE III
COMPUTATIONAL OPERATION COUNTS FOR EACH ITERATION IN NMF AND NLCF

	F-norm Formulation			
	Fladd	Flmlt	Fldiv	Overall
NMF	$2MNK + 2(M+N)K^2$	$2MNK + 2(M+N)K^2 + (M+N)K$	$(M+N)K$	$O(MNK)$
NLCF	$2MNK + (3M+2N)K^2 + MN + 2MK + 3NK$	$2MNK + (3M+2N)K^2 + MN + 4MK + 4NK$	$(M+N)K$	$O(MNK)$

N: the number of sample points M: the number of features K: the number of factors

There is no difficulty in counting the computational operation counts for each matrices multiplication, and it is presented in Table II. The computational operation counts of $\mathbf{C}$ and $\mathbf{D}$ need more explanation. From the definition of $\mathbf{C}$, we need to compute $\mathbf{X}^T\mathbf{X}$, which costs N^2M flam. In reality, there is no need to do this matrices multiplication, we only need its diagonal entries. Note that $\mathbf{c} = \sum_{j=1}^{M} x_{ji}^2$, where $i = 1, \ldots, N$. So it only costs NM flam to compute C. Similarly, computation of D need MK flam.

We also summarize the computational operation counts for each iteration of NLCF in Table III. Suppose the multiplicative updates stop after t iterations, the overall cost for NMF (F-norm formulation) and NLCF are both $O(tMNK)$.

E. Incorporate Geometrically Based Regularizer

Another limitation of NMF is that it fails to discover the intrinsic geometrical and discriminating structure of the data space, which is essential to the real-world applications [32], [33]. Cai et al. have proposed a new version of NMF called Graph regularized Non-negative Matrix Factorization [18], [20], [23], which adds a geometrically based regularizer to the original NMF Objective function. In this section, we will incorporate this regularizer to out NLCF algorithm.

The geometrically based regularizer is the following one:

$$\mathcal{R} = \mathrm{Tr}(\mathbf{VLV}^T). \tag{9}$$

$\mathrm{Tr}(\cdot)$ denotes the trace of a matrix. $\mathbf{L} = \mathbf{E} - \mathbf{W}$ is called graph Laplacian, where $\mathbf{W}$ is the weight matrix and the $\mathbf{W}_{ij}$ is used to measure the closeness of two points $\mathbf{x}_i$ and $\mathbf{x}_j$ (please see [23] for the details of how to build weight matrix). $\mathbf{E}$ is a diagonal matrix whose entries are column (or row, since $\mathbf{W}$ is symmetric) sums of $\mathbf{W}$.

Now we add this regularizer to our NLCF Objective function,

$$\mathcal{O} = \|\mathbf{X} - \mathbf{UV}\|^2 + \sum_{i=1}^{N} \mu \sum_{k=1}^{K} |v_{ki}| \, \|\mathbf{u}_k - \mathbf{x}_i\|^2 + \lambda \mathrm{Tr}(\mathbf{VLV}^T). \tag{10}$$

From the updating rules of GNMF [23], we have the following updating rules:

$$u_{jk} \leftarrow u_{jk} \frac{(\mathbf{XV}^T + \mu \sum_{i=1}^{N} \mathbf{x}_i \mathbf{1}^T \Lambda_i)_{jk}}{(\mathbf{UVV}^T + \mu \sum_{i=1}^{N} \mathbf{U}\Lambda_i)_{jk}} \tag{11}$$

$$v_{ki} \leftarrow v_{ki} \frac{2((\mu+1)(\mathbf{U}^T\mathbf{X}) + \lambda \mathbf{VW})_{ki}}{(2\mathbf{U}^T\mathbf{UV} + \mu\mathbf{C} + \mu\mathbf{D} + 2\lambda\mathbf{VE})_{ki}} \tag{12}$$

TABLE IV
STATISTICS OF THREE DATA SETS

Dataset	Size (N)	Dimensionality (M)	# of Classes (K)
ORL	400	1024	40
MNIST	4000	784	10
Yale	165	1024	15

we will guarantee that the update rules of $\mathbf{U}$ and $\mathbf{V}$ in Eq. (11) and Eq. (12) converge and the final solution will be a local optimum. Please see the Appendix for a detailed proof.

IV. EXPERIMENTAL RESULTS

In this section, various experiments are performed to demonstrate the effectiveness of our proposed Non-negative Local Coordinate Factorization method.

A. Data Corpora

Three image data sets are used in the experiment. The important statistics of these data sets are summarized below (see also Table IV):

1) The first one is Cambridge ORL face database[1]. There are ten different images of each of 40 distinct subjects. All the images were taken against a dark homogeneous background with the subjects in an upright, frontal position. We crop the original 112×92 images into 64×64 gray scale images.
2) The second one is the MNIST database of handwritten digits[2]. We use a test set of 4,000 examples for clustering, which contains 28×28 gray scale images of 10 digits.
3) The third one is the Yale Face Database[3] containing 32×32 gray scale images of 15 individuals. There are 11 images per subject facial expression or configuration.

B. Clustering Evaluation

Previous studies show that NMF is very powerful for data clustering. It is superior to the Latent Semantic Indexing method (LSI) [34] and several popular spectral clustering methods [35]. In this experiment, we investigate the effectiveness of our proposed algorithm on image clustering.

We set the parameter K to be the number of clusters and use the obtained coefficient matrix $\mathbf{V}$ to determine the cluster

[1] http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html.
[2] http://yann.lecun.com/exdb/mnist/.
[3] http://cvc.yale.edu/projects/yalefaces/yalefaces.html.

TABLE V
CLUSTERING PERFORMANCE ON ORL DATABASE

Cluster Number	Accuracy (Mean ± Std%)					Normalized Mutual Information (Mean ± Std%)				
	NMF	NLCF	NLCF-G	Kmeans	NMF-SC	NMF	NLCF	NLCF-G	Kmeans	NMF-SC
2	98.3 ± 3.2	**98.5 ± 3.2**	98.5 ± 4.5	92.5 ± 10.6	98.0 ± 2.0	92.9 ± 15.1	93.7 ± 13.1	94.9 ± 15.2	76.6 ± 31.7	**95.2 ± 9.7**
4	79.3 ± 15.0	82.0 ± 12.7	**83.5 ± 12.7**	72.0 ± 13.7	81.0 ± 13.4	70.6 ± 17.9	75.5 ± 15.9	**78.3 ± 15.4**	64.7 ± 15.1	75.2 ± 17.2
8	62.3 ± 5.5	**76.6 ± 13.2**	69.3 ± 9.3	62.1 ± 7.5	72.5 ± 8.7	64.4 ± 5.2	**78.8 ± 12.0**	73.5 ± 10.0	67.6 ± 9.4	73.1 ± 6.8
12	56.3 ± 3.4	**70.8 ± 7.3**	65.0 ± 4.2	55.3 ± 9.4	63.0 ± 6.6	64.1 ± 2.9	**77.3 ± 5.0**	73.7 ± 3.4	66.4 ± 9.4	70.0 ± 4.8
16	53.5 ± 3.4	**67.4 ± 4.8**	63.5 ± 7.5	57.5 ± 6.8	58.3 ± 3.7	65.0 ± 3.0	**77.1 ± 2.9**	74.2 ± 6.5	70.9 ± 5.5	68.1 ± 2.5
20	48.1 ± 4.5	**64.7 ± 7.2**	60.4 ± 5.1	55.4 ± 4.0	50.8 ± 3.2	62.4 ± 3.0	**75.4 ± 3.7**	73.4 ± 3.4	70.0 ± 3.4	64.3 ± 1.7
25	46.0 ± 3.2	**63.5 ± 2.1**	57.4 ± 2.8	56.7 ± 1.5	50.2 ± 3.8	62.9 ± 2.1	**76.5 ± 1.7**	72.4 ± 2.4	72.5 ± 1.9	65.3 ± 2.6
30	41.9 ± 3.1	**60.4 ± 3.1**	54.6 ± 3.1	52.3 ± 2.6	45.7 ± 2.1	61.2 ± 2.1	**75.4 ± 2.4**	72.4 ± 2.6	70.6 ± 2.1	63.4 ± 1.6
40	39.5	**61.8**	53.8	47.5	41.0	61.6	**76.5**	73.4	68.7	61.4
Avg	58.4	**71.7**	67.3	61.3	62.3	67.2	**78.5**	76.3	69.8	70.7

TABLE VI
CLUSTERING PERFORMANCE ON MNIST DATABASE

Cluster Number	Accuracy (Mean ± Std%)					Normalized Mutual Information (Mean ± Std%)				
	NMF	NLCF	NLCF-G	Kmeans	NMF-SC	NMF	NLCF	NLCF-G	Kmeans	NMF-SC
2	88.0 ± 13.5	89.0 ± 14.5	**89.1 ± 14.8**	88.8 ± 13.7	85.4 ± 14.2	58.8 ± 29.0	**63.2 ± 30.0**	63.8 ± 30.6	61.8 ± 29.2	51.0 ± 28.7
3	78.4 ± 11.5	82.9 ± 9.2	**84.2 ± 9.5**	77.1 ± 13.5	82.1 ± 8.8	52.0 ± 12.0	59.0 ± 11.2	**60.9 ± 11.9**	54.6 ± 13.1	57.1 ± 10.7
4	75.6 ± 10.5	**82.0 ± 8.6**	79.9 ± 10.6	73.2 ± 10.1	74.4 ± 11.9	51.7 ± 12.6	**60.5 ± 10.8**	59.6 ± 10.8	55.4 ± 8.9	51.7 ± 12.9
5	63.5 ± 10.0	**71.5 ± 10.3**	70.8 ± 11.6	68.0 ± 7.8	62.4 ± 7.2	45.7 ± 10.5	55.6 ± 9.5	**56.4 ± 10.8**	52.9 ± 7.0	45.2 ± 7.0
6	52.9 ± 4.3	**63.9 ± 6.0**	62.8 ± 4.3	58.7 ± 6.4	57.4 ± 3.7	40.0 ± 4.4	**50.3 ± 5.7**	49.6 ± 5.7	48.2 ± 7.5	43.0 ± 3.0
7	52.8 ± 5.9	**63.1 ± 8.1**	61.5 ± 8.8	60.9 ± 10.6	56.1 ± 5.2	40.6 ± 4.4	49.2 ± 5.3	**51.2 ± 5.0**	49.2 ± 7.1	42.9 ± 4.4
8	51.4 ± 5.5	**62.4 ± 5.8**	59.8 ± 6.8	58.1 ± 5.5	56.6 ± 5.2	40.8 ± 4.6	**50.4 ± 3.4**	50.3 ± 3.8	48.3 ± 3.3	44.6 ± 4.0
9	45.3 ± 2.7	53.9 ± 4.2	**55.9 ± 4.7**	52.0 ± 4.0	46.2 ± 4.0	36.8 ± 1.7	46.6 ± 2.4	**47.7 ± 2.5**	46.3 ± 1.9	37.0 ± 3.8
10	43.8	**57.0**	55.0	50.4	48.0	35.6	47.0	**48.1**	45.4	41.5
Avg	61.3	**69.5**	68.8	65.2	63.2	44.7	53.5	**54.2**	51.3	46.0

label of each data point. More precisely, we examine each column of the matrix $\mathbf{V}$, and assign the sample $\mathbf{x}_i$ to cluster c if $c = \arg\max_k v_{ki}$.

1) Evaluation Metric: The clustering result is evaluated by comparing the obtained label of each sample with that provided by the data set. Three metrics have been used in our experiments. The accuracy (AC) [36] and the normalized mutual information metric ($\overline{MI}$) [36] are used to measure the clustering performance, while sparseness (SP) [11] measures the sparseness of coefficients matrix.

Given a data point $\mathbf{x}_i$, let r_i and s_i be the cluster label and the label provided by the corpus, respectively. The AC is defined as follows:

$$AC = \frac{\sum_{i=1}^{N} \delta(s_i, map(r_i))}{N}$$

where N is the total number of samples and $\delta(x, y)$ is the delta function that equals one if $x = y$ and equals zero otherwise, and $map(r_i)$ is the permutation mapping function that maps each cluster label r_i to the equivalent label from the data corpus. The best mapping can be found by using the Kuhn-Munkres algorithm [37].

On the other hand, let C denote the set of clusters obtained from the ground truth and C' obtained from our algorithm. Their mutual information metric $MI(C, C')$ is defined as follows:

$$MI(C, C') = \sum_{c_i \in C, c'_j \in C'} p(c_i, c'_j) \cdot log_2 \frac{p(c_i, c'_j)}{p(c_i) \cdot p(c'_j)}$$

where $p(c_i)$ and $p(c'_j)$ are the probabilities that a sample arbitrarily selected from the data set belongs to the clusters c_i and c'_j, respectively, and $p(c_i, c'_j)$ is the joint probability that the arbitrarily selected sample belongs to the clusters c_i as well as c'_j at the same time. In our experiments, we use the normalized mutual information $\overline{MI}$ as follows:

$$\overline{MI}(C, C') = \frac{MI(C, C')}{max(H(C), H(C'))}$$

where $H(C)$ and $H(C')$ are the entropies of C and C', respectively. It is easy to check that $\overline{MI}(C, C')$ ranges from 0 to 1. $\overline{MI} = 1$ if the two sets of clusters are identical, and $\overline{MI} = 0$ if the two sets are independent.

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

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

where $\mathbf{v}$ is a column vector of $\mathbf{V}$. If all components of $\mathbf{v}$ are equal, $SP(\mathbf{v})$ evaluates to unity, and if $\mathbf{v}$ only contains a single non-zero component, $SP(\mathbf{v})$ evaluates to zero. In our experiments, we take the average sparseness over all the new representations (column vectors of $\mathbf{V}$).

2) Clustering Results: To show the improvement of the clustering performance by our method, we compared NLCF and NLCF-G(NLCF with Graph regularizer) with the following three popular algorithms:

1) Non-negative Matrix Factorization based clustering (NMF in short).

TABLE VII
CLUSTERING PERFORMANCE ON YALE DATABASE

Cluster Number	Accuracy (Mean ± Std%)					Normalized Mutual Information (Mean ± Std%)				
	NMF	NLCF	NLCF-G	Kmeans	NMF-SC	NMF	NLCF	NLCF-G	Kmeans	NMF-SC
2	72.3 ± 11.2	**85.0 ± 11.0**	84.1 ± 10.6	71.4 ± 11.3	65.5 ± 8.9	20.6 ± 14.9	**48.1 ± 23.7**	44.1 ± 22.4	21.1 ± 18.6	15.1 ± 12.6
3	60.6 ± 8.1	68.5 ± 15.3	**69.7 ± 17.9**	60.9 ± 12.1	66.7 ± 15.4	31.1 ± 12.3	43.8 ± 22.5	**44.7 ± 24.9**	30.2 ± 16.6	34.5 ± 20.6
4	61.4 ± 9.7	**63.9 ± 8.7**	63.4 ± 8.5	51.1 ± 7.6	59.8 ± 8.9	43.8 ± 12.7	**46.3 ± 12.3**	**46.3 ± 12.3**	33.0 ± 9.5	40.3 ± 11.0
5	52.0 ± 5.0	**55.6 ± 9.1**	54.7 ± 7.9	46.6 ± 11.4	54.6 ± 4.5	36.3 ± 8.6	**40.0 ± 11.3**	39.0 ± 10.1	28.3 ± 11.9	39.7 ± 6.7
6	51.4 ± 10.1	52.6 ± 8.1	52.3 ± 10.2	47.1 ± 6.8	**54.9 ± 6.7**	40.6 ± 11.5	42.3 ± 10.6	**43.9 ± 11.6**	37.5 ± 5.7	43.6 ± 6.9
7	50.8 ± 6.9	50.8 ± 5.9	**51.6 ± 8.9**	48.2 ± 7.9	50.0 ± 4.5	41.9 ± 6.6	43.5 ± 7.1	**44.2 ± 9.0**	40.9 ± 9.8	42.1 ± 6.1
8	43.0 ± 4.9	**49.6 ± 5.6**	46.5 ± 5.9	46.0 ± 7.1	47.3 ± 4.9	38.0 ± 5.2	**43.2 ± 7.0**	41.4 ± 6.7	40.7 ± 8.3	42.6 ± 6.1
9	43.8 ± 4.0	**47.5 ± 4.8**	47.2 ± 3.8	41.5 ± 5.2	45.6 ± 7.0	40.4 ± 4.2	43.0 ± 4.4	**45.6 ± 4.2**	40.1 ± 6.3	41.8 ± 7.2
10	40.2 ± 3.6	**48.3 ± 4.9**	47.7 ± 5.7	40.7 ± 7.4	45.8 ± 4.9	39.5 ± 2.6	47.0 ± 4.6	**47.5 ± 5.0**	41.5 ± 6.3	45.2 ± 4.2
11	39.7 ± 3.4	**46.0 ± 3.4**	45.5 ± 4.9	42.3 ± 4.2	43.6 ± 5.2	40.2 ± 2.2	46.9 ± 2.6	**47.0 ± 4.1**	41.9 ± 5.3	45.0 ± 4.6
12	36.7 ± 2.7	43.4 ± 4.0	42.1 ± 3.1	38.9 ± 5.2	**44.6 ± 4.3**	38.7 ± 2.2	46.1 ± 3.2	45.8 ± 2.0	40.5 ± 4.3	**46.8 ± 4.0**
13	37.0 ± 3.6	**45.0 ± 2.4**	42.2 ± 3.5	42.2 ± 3.7	44.8 ± 2.0	41.0 ± 3.2	**49.0 ± 2.5**	47.9 ± 2.5	45.6 ± 4.0	48.3 ± 2.7
14	35.0 ± 2.0	**42.3 ± 3.0**	41.6 ± 2.7	37.7 ± 4.5	42.1 ± 2.8	40.3 ± 1.7	47.4 ± 2.7	**47.8 ± 2.1**	42.8 ± 4.0	46.5 ± 1.6
15	36.4	**49.1**	42.3	37.0	40.0	41.0	**53.7**	48.5	40.2	48.8
Avg	47.2	**53.4**	52.2	46.5	50.4	38.1	**45.7**	45.3	37.5	41.5

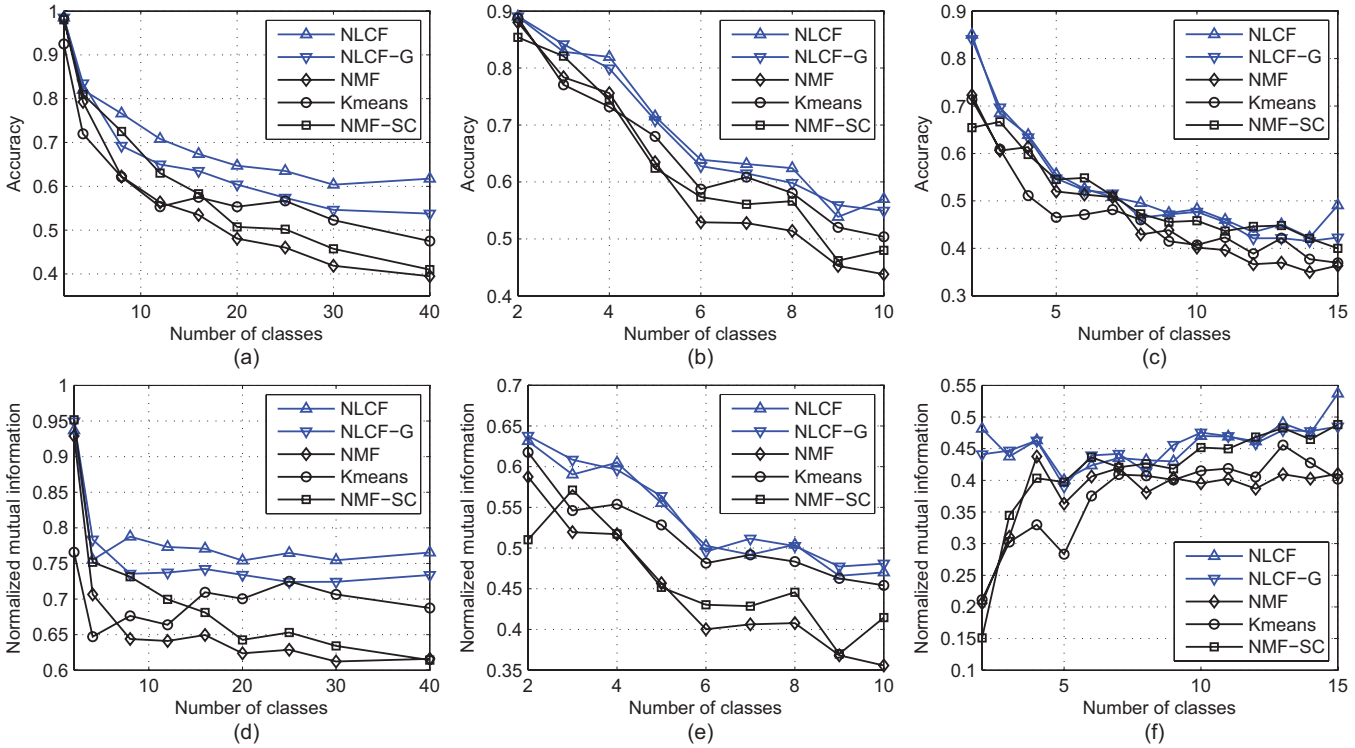

Fig. 1. Accuracy versus the number of clusters on (a) ORL, (b) MNIST, and (c) Yale databases. The normalized mutual information versus the number of clusters on (d) ORL, (e) MNIST, and (f) Yale databases.

2) Canonical K-means clustering method (Kmeans in short).
3) Non-negative Matrix Factorization with Sparseness Constraints (NMF-SC in short, [11]).

The evaluations were conducted with different numbers of clusters. On ORL data set, the cluster number ranges from 2 to 40. On MNIST data set, the cluster number ranges from 2 to 10. On Yale data set, the cluster number ranges from 2 to 15. For each given cluster number, 10 test runs were conducted on different randomly chosen clusters. The final performance is recorded by averaging the performance of the 10 tests.

Table V, VI, VII and Fig. 1 show the clustering results on the data sets ORL, MNIST and Yale. The average sparseness of the coefficients matrix is reported in Table VIII.

On ORL data set, the average clustering accuracies obtained by NLCF, NLCF-G, NMF, NMF-SC, and Kmeans are 71.7%, 67.3%, 58.4%, 62.3%, and 61.3%, respectively. Comparing to the third best approach, that is, NMF-SC, NLCF achieves 9.4% accuracy improvement and NLCF-G achieves 5.0%. For mutual information, it can be seen that NLCF achieves 7.8%, NLCF-G achieves 5.6% improvement over NMF-SC. On MNIST data set, the average clustering accuracies obtained by NLCF, NLCF-G, NMF, NMF-SC, and Kmeans are 69.5%, 68.8%, 61.3%, 63.2%, and 65.2%, respectively. Again, NLCF and NLCF-G significantly outperform other three algorithms in terms of both accuracy and mutual information. On Yale data set, the average clustering accuracies obtained by NLCF, NLCF-G, NMF, NMF-SC and Kmeans are 53.4%,

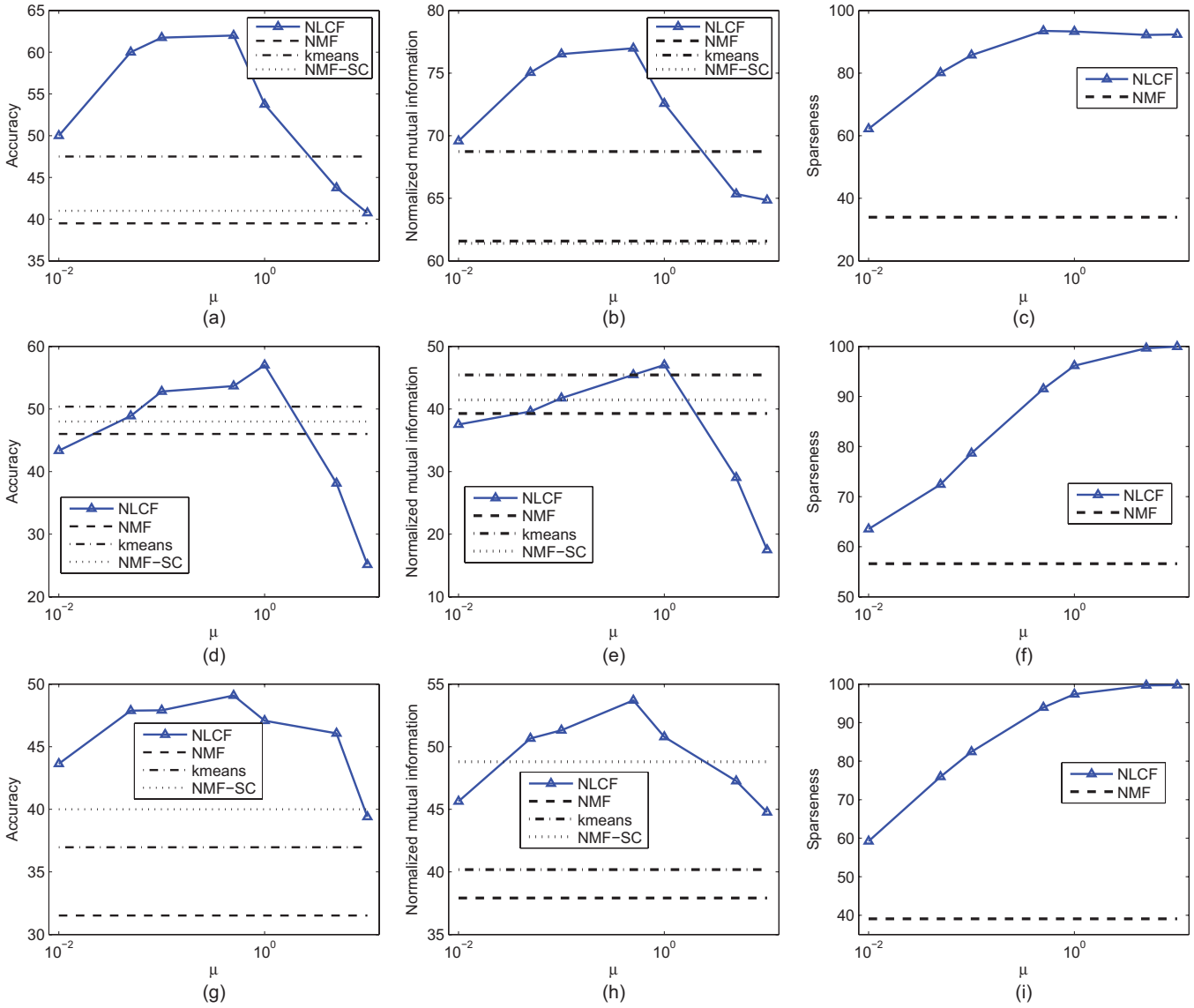

Fig. 2. Accuracy of NLCF versus the parameter μ on (a) ORL, (d) MNIST, and (g) Yale databases. The normalized mutual information of NLCF versus the parameter μ on (b) ORL, (e) MNIST, and (h) Yale databases. The sparseness of NLCF versus the parameter μ on (c) ORL, (f) MNIST, and (i) Yale databases.

TABLE VIII
AVERAGE SPARSENESS OF COEFFICIENTS MATRIX

Database	Sparseness (%)	
	NMF	NLCF
ORL	34.4	84.3
MNIST	59.3	96.5
Yale	40.5	93.3

52.2%, 47.2%, 50.4% and 46.5%, respectively. In this data set, NLCF, NLCF-G and NMF-SC get similar performance and are superior to NMF and Kmeans. But NLCF and NLCF-G still narrowly beat NMF-SC both in accuracy and mutual information.

The sparseness of the encodings obtained by NLCF is greater than 80 on both data sets. This indicates that our proposed approach can indeed obtain highly sparse representations, which in turn, improves the clustering performance.

3) Parameter Selection: Our NLCF model has only one essential parameter: the regularization parameter μ. NLCF boils down to the original NMF when the regularization parameter $\mu = 0$. As μ increases, we expect the learned encodings become more sparse.

Fig. 2 shows how the average clustering performance and the sparseness of learned encodings vary with the parameters μ, respectively. As we can see, NLCF achieves good performance with the μ varying from 0.1 to 1, and the sparseness of the encodings increases as μ increases.

C. Basis Vectors and Image Encodings

In this test, we randomly select 25 subjects from the ORL database and for each subject we randomly select 5 face images. Fig. 3 shows the sample images from the ORL database, and the basis vectors and image encodings obtained by NMF and NLCF.

Comparing the basis images obtained by NLCF with the original face images, we find that the basis images look like the original face images very much. This shows that our local

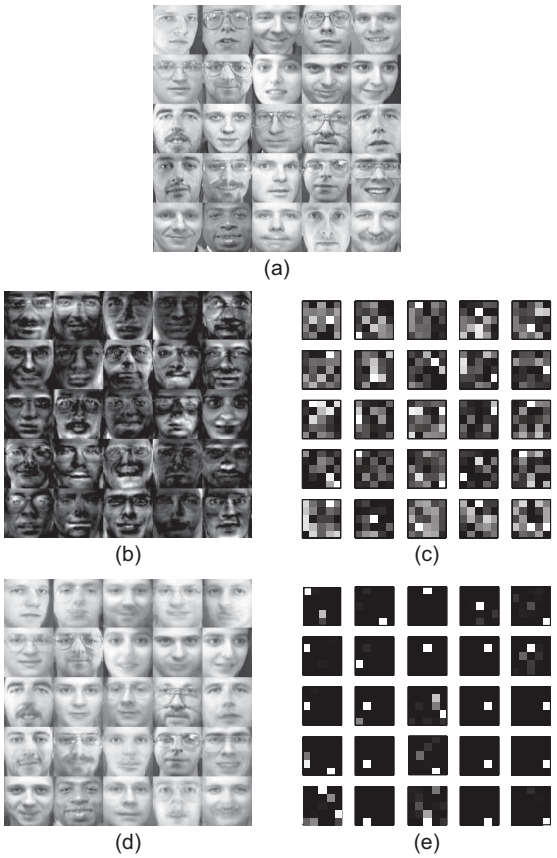

Fig. 3. 5×5 montages. (a) Original images. (b) and (d) Basis vectors learned by NMF and NLCF. (c) and (e) Image encodings (i.e., the obtained new representations) of the two methods. Positive values are illustrated with white pixels. The encodings learned by NLCF are much more sparse than those learned by NMF.

coordinate constraint can indeed generate basis images (i.e. the anchor points) which are sufficiently close to the original images.

Comparing with the image encodings obtained by NMF, the image encodings obtained by NLCF are much more sparse. For NLCF, more than half of the image encodings only have one nonzero element. And the nonzero element is exactly the coordinate coefficient with respect to the basis image which is closest to this face image.

D. Learning Overcomplete Basis

Usually, the parameter K (the dimension of the new representations) are set to be less than the dimension of the original data space. However, in some cases, it is desirable to learn overcomplete basis [38]–[41], where K is set to be larger than the original dimension. This problem has received considerable attention since the work of Olshausen and Field [39], who suggest that this is the strategy used by the visual cortex for representing images. The implication is that a sparse, overcomplete representation is special suitable for visual tasks such as object detection and recognition that occur in higher regions of the cortex [40]. We give a simple synthetic example to show how our proposed algorithm performs for learning overcomplete basis.

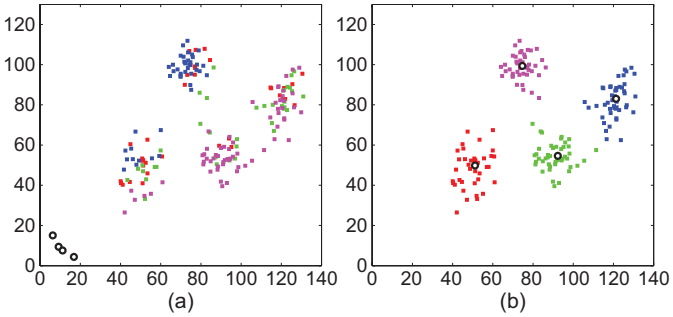

Fig. 4. Toy example of learning overcomplete basis. The black circles denote the learned basis vectors. Each color represents a cluster obtained by (a) NMF or (b) NLCF.

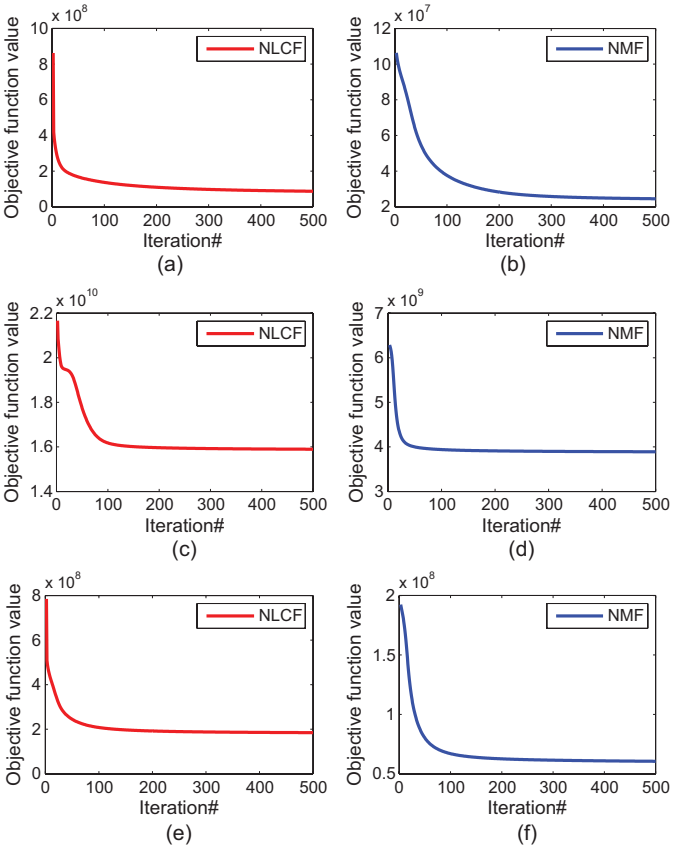

Fig. 5. Convergence curves of NLCF on (a) ORL, (c) MNIST, and (e) Yale databases. The convergence curves of NMF on (b) ORL, (d) MNIST, and (f) Yale databases.

We randomly generate 180 points from mixture of four Gaussians in a 2-dimensional space. NMF and NLCF are performed to cluster these data points into four clusters, as shown in Fig. 4. As we can see, NLCF performs much better than NMF. Note that the dimension of the input data is 2, but we use 4 basis vectors. The four basis vector obtained by NLCF exactly reside at the centers of the four clusters, one for each. However, the basis vector obtained by NMF are far away from the data points. The reason is that the four basis vectors (in fact, two are sufficient) span the two-dimensional space. Thus, there will be infinitely many solutions for NMF leading to zero reconstruction error. For our algorithm, it introduces a local coordinate constraint which require the basis vector to be sufficiently close to the data points.

E. Convergence Study

We have proved that the updating rules for minimizing the objective function of NLCF are convergent. Here we investigate how fast the algorithm can converge and compare with NMF.

Fig. 5 shows the convergence curves of NLCF on the three data sets. For each figure, the y-axis is the value of objective function and the x-axis is the iteration number. As can be seen, NLCF converges within 20 iterations on the ORL database, within 100 iterations on the MNIST database and within 50 iterations on the Yale database. NLCF converges faster than NMF on both ORL and Yale databases but slower on the MNIST database.

V. Conclusion

We have presented a novel method for matrix factorization, called Non-negative Local Coordinate Factorization (NLCF). NLCF aims to ensure sparseness of the new representations by adding a local coordinate constraint. The learned basis vector are close to the cluster centers. Thus, each data point can be represented by linear combination of only few basis vectors, yielding sparse representation. This property also makes the algorithm particularly suitable for data clustering, as demonstrated in our experiments. We have also shown that NLCF is more effective than NMF for learning overcomplete basis.

One question remains to be investigated in our future work: There is another objective function of NMF, the "divergence" one. How to incorporate the local coordinate constraint into the divergence objective function is a remaining problem.

Appendix
Proof of Convergence

In this section, we show that the iteration steps in Eq. (7), Eq. (8) and the iteration steps in Eq. (11), Eq. (12) are convergent. As we know that Eq. (10) is a general version of Eq. (3), if we set the parameter λ to zero, Eq. (3) boils down to Eq. (10). so we just give the proof of the general version.

We have the following theorem:

Theorem 1: The objective function $\mathcal{O}$ in Eq. (10) is nonincreasing under the update rules in Eq. (11) and Eq. (12). The objective function is invariant under these updates if and only if $\mathbf{U}$ and $\mathbf{V}$ are at a stationary point.

To prove Theorem 1, we need to show that the objective function Eq. (10) is bounded from below and nonincreasing under the update steps in Eq. (11) and Eq. (12). Since the objective function $\mathcal{O}$ is greater than zero, we only need to verify that $\mathcal{O}$ is nonincreasing under the update steps in Eq. (11) and Eq. (12).

Our proof will make use of an auxiliary function similar to that used in the Expectation-Maximization algorithm [42]: *definition:* $G(u, u')$ is an auxiliary function for $F(u)$ if the conditions

$$G(u, u') \geq F(u), G(u, u) = F(u)$$

are satisfied.

The auxiliary function is a useful concept because of the following lemma:

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

$$u^{(t+1)} = \arg\min_{u} G(u, u^{(t)}) \tag{13}$$

Proof:

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

■

Now we will show that the update step for $\mathbf{U}$ in Eq. (11) is exactly the update in Eq. (13) with a proper auxiliary function. We rewrite the objective function $\mathcal{O}$ in Eq. (10) as follows

$$\mathcal{O} = \sum_{j=1}^{M}\sum_{i=1}^{N}(x_{ji} - \sum_{k=1}^{K} u_{jk}v_{ki})^2 + \mu \sum_{i=1}^{N}\Big(\sum_{k=1}^{K}|v_{ki}|\sum_{j=1}^{M}(u_{jk} - x_{ji})^2\Big)$$

Considering any element u_{ab} in U, we use F_{ab} to denote the part of $\mathcal{O}$ which is only relevant to u_{ab}. From Eq. (4), it is easy to check that

$$F'_{ab} = \Big(\frac{\partial \mathcal{O}}{\partial \mathbf{U}}\Big)_{ab} = (2\mathbf{U}\mathbf{V}\mathbf{V}^T - 2\mathbf{X}\mathbf{V}^T)_{ab} + \mu\sum_{i=1}^{N}(-2\mathbf{x}_i\mathbf{1}^T\Lambda_i + 2\mathbf{U}\Lambda_i)_{ab}$$

$$F''_{ab} = 2(\mathbf{V}\mathbf{V}^T)_{bb} + 2\mu\sum_{i=1}^{N}(\Lambda_i)_{bb}$$

Since our update is essentially element-wise, it is sufficient to show that each F_{ab} is nonincreasing under the update step of Eq. (11).

Lemma 2: The function

$$G(u, u_{ab}^{(t)}) = F_{ab}(u_{ab}^{(t)}) + F'_{ab}(u_{ab}^{(t)})(u - u_{ab}^{(t)}) + \frac{(\mathbf{U}\mathbf{V}\mathbf{V}^T)_{ab} + \mu\sum_{i=1}^{N}(\mathbf{U}\Lambda_i)_{ab}}{u_{ab}^{(t)}}(u - u_{ab}^{(t)})^2 \tag{14}$$

is an auxiliary function for F_{ab}. The part of $\mathcal{O}$ which is only relevant to u_{ab}.

Proof: Since $G(u, u) = F_{ab}(u)$ is obvious, we only need to show that $G(u, u_{ab}^{(t)}) \geq F_{ab}(u)$. To do this, we compare the Taylor series expansion of $F_{ab}(u)$

$$F_{ab}(u) = F_{ab}(u_{ab}^{(t)}) + F'_{ab}(u_{ab}^{(t)})(u - u_{ab}^{(t)}) + \Big((\mathbf{V}\mathbf{V}^T)_{bb} + \mu\sum_{i=1}^{N}(\Lambda_i)_{bb}\Big)(u - u_{ab}^{(t)})^2$$

with Eq. (14) to find that $G(u, u_{ab}^{(t)}) \geq F_{ab}(u)$ is equivalent to

$$(\mathbf{U}\mathbf{V}\mathbf{V}^T)_{ab} + \mu\sum_{i=1}^{N}(\mathbf{U}\Lambda_i)_{ab} \geq (\mathbf{V}\mathbf{V}^T)_{bb}u_{ab}^{(t)} + \mu u_{ab}^{(t)}\sum_{i=1}^{N}(\Lambda_i)_{bb} \tag{15}$$

We have

$$(\mathbf{UVV}^T)_{ab} = \sum_{l=1}^{K} u_{al}^{(t)}(\mathbf{VV}^T)_{lb} \geq (\mathbf{VV}^T)_{bb}u_{ab}^{(t)}$$

and

$$\begin{aligned}\mu\sum_{i=1}^{N}(\mathbf{U}\Lambda_i)_{ab} &= \mu\sum_{i=1}^{N}\Big(\sum_{l=1}^{K}u_{al}^{(t)}(\Lambda_i)_{lb}\Big)\\ &= \mu\sum_{i=1}^{N}\Big(u_{ab}^{(t)}(\Lambda_i)_{bb}\Big)\\ &= \mu u_{ab}^{(t)}\sum_{i=1}^{N}(\Lambda_i)_{bb}\end{aligned}$$

Thus, Eq. (15) holds and $G(u, u_{ab}^{(t)}) \geq F_{ab}(u)$. ■

Then we are going to show that the update step for V in Eq. (12) is exactly the update in Eq. (13) with a proper auxiliary function just like $\mathbf{U}$.

Considering any element v_{ab} in V, we still use H_{ab} to denote the part of $\mathcal{O}$ which is only relevant to v_{ab}. It is easy to check that

$$\begin{aligned}H'_{ab} &= \Big(\frac{\partial\mathcal{O}}{\partial\mathbf{V}}\Big)_{ab}\\ &= (2\mathbf{U}^T\mathbf{UV} - 2\mathbf{U}^T\mathbf{X} + \mu(\mathbf{C} - 2\mathbf{U}^T\mathbf{X} + \mathbf{D}) + 2\lambda\mathbf{VL})_{ab}\\ H''_{ab} &= 2(\mathbf{U}^T\mathbf{U})_{aa} + 2\lambda\mathbf{L}_{bb}.\end{aligned}$$

Now we have the following lemma.

Lemma 3: Function

$$\begin{aligned}G(v, v_{ab}^{(t)}) = H_{ab}(v_{ab}^{(t)}) + H'_{ab}(v_{ab}^{(t)})(v - v_{ab}^{(t)})\\ + \frac{(\mathbf{U}^T\mathbf{UV} + \frac{1}{2}\mu\mathbf{C} + \frac{1}{2}\mu\mathbf{D} + \lambda\mathbf{VE})_{ab}}{v_{ab}^{(t)}}(v - v_{ab}^{(t)})^2\end{aligned} \tag{16}$$

is an auxiliary function for H_{ab}, the part of $\mathcal{O}$ which is only relevant to v_{ab}.

Proof: Since $G(v, v) = H_{ab}(v)$, we only need to show that $G(v, v_{ab}^{(t)}) \geq H_{ab}(v)$. We compare the Taylor series expansion of $H_{ab}(v)$

$$\begin{aligned}H_{ab}(v) = H_{ab}(v_{ab}^{(t)}) + H'_{ab}(v_{ab}^{(t)})(v - v_{ab}^{(t)})\\ +((\mathbf{U}^T\mathbf{U})_{aa} + \lambda\mathbf{L}_{bb})(v - v_{ab}^{(t)})^2\end{aligned}$$

with Eq. (16) to find that $G(v, v_{ab}^{(t)}) \geq H_{ab}(v)$ is equivalent to

$$(\mathbf{U}^T\mathbf{UV} + \frac{1}{2}\mu\mathbf{C} + \frac{1}{2}\mu\mathbf{D} + \lambda\mathbf{VE})_{ab} \geq v_{ab}^{(t)}((\mathbf{U}^T\mathbf{U})_{aa} + \lambda\mathbf{L}_{bb}) \tag{17}$$

We have

$$(\mathbf{U}^T\mathbf{UV})_{ab} = \sum_{l=1}^{K}(\mathbf{U}^T\mathbf{U})_{al}v_{lb}^{(t)} \geq v_{ab}^{(t)}(\mathbf{U}^T\mathbf{U})_{aa}$$

and

$$\begin{aligned}\lambda(\mathbf{VE})_{ab} &= \sum_{i=1}^{N}v_{ai}^{(t)}\mathbf{E}_{ib} \geq \lambda v_{ab}^{(t)}\mathbf{E}_{bb}\\ &\geq \lambda v_{ab}^{(t)}(\mathbf{E} - \mathbf{W})_{bb} = \lambda v_{ab}^{(t)}\mathbf{L}_{bb}\end{aligned}$$

Thus, Eq. (17) holds and $G(v, v_{ab}^{(t)}) \geq H_{ab}(v)$. ■

We can now demonstrate the convergence of Theorem 1:

Proof of Theorem 1: Replacing $G(u, u_{ab}^{(t)})$ in Eq. (13) by Eq. (14) and replacing $G(v, v_{ab}^{(t)})$ in Eq. (13) by Eq. (16) results in the update rule:

$$\begin{aligned}u_{ab}^{(t+1)} &= u_{ab}^{(t)} - u_{ab}^{(t)}\frac{F'_{ab}(u_{ab}^{(t)})}{2(\mathbf{UVV}^T)_{ab} + 2\mu\sum_{i=1}^{N}(\mathbf{U}\Lambda_i)_{ab}}\\ &= u_{ab}^{(t)}\frac{(\mathbf{XV}^T + \mu\sum_{i=1}^{N}\mathbf{x}_i\mathbf{1}^T\Lambda_i)_{ab}}{(\mathbf{UVV}^T + \mu\sum_{i=1}^{N}\mathbf{U}\Lambda_i)_{ab}}\\ v_{ab}^{(t+1)} &= v_{ab}^{(t)} - v_{ab}^{(t)}\frac{H'_{ab}(v_{ab}^{(t)})}{(2\mathbf{U}^T\mathbf{UV} + \mu\mathbf{C} + \mu\mathbf{D} + 2\lambda\mathbf{VE})_{ab}}\\ &= v_{ab}^{(t)}\frac{2(\mu+1)(\mathbf{U}^T\mathbf{X})_{ab}}{(2\mathbf{U}^T\mathbf{UV} + \mu\mathbf{C} + \mu\mathbf{D} + 2\lambda\mathbf{VE})_{ab}}\end{aligned}$$

Since Eq. (14) and Eq. (16) are auxiliary functions, F_{ab} and H_{ab} are nonincreasing under the update rule.

Theorem 1 guarantees that the update rules of $\mathbf{U}$ and $\mathbf{V}$ in Eq. (11), Eq. (12) converge and the final solution will be a local optimum.

REFERENCES

[1] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification*, 2nd ed. New York: Wiley, 2000.

[2] D. D. Lee and H. S. Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, pp. 788–791, Oct. 1999.

[3] X. Li, Y. Pang, and Y. Yuan, "L1-norm-based 2DPCA," *IEEE Trans. Syst., Man, Cybern. B, Cybern.*, vol. 40, no. 4, pp. 1170–1175, Aug. 2010.

[4] Y. Pang, X. Li, and Y. Yuan, "Robust tensor analysis with L1-norm," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 20, no. 2, pp. 172–178, Feb. 2010.

[5] C. Wu, J. Zhu, D. Cai, C. Chen, and J. Bu, "Semi-supervised nonlinear hashing using bootstrap sequential projection learning," *IEEE Trans. Knowl. Data Eng.*, Mar. 2012.

[6] N. K. Logothetis and D. L. Sheinberg, "Visual object recognition," *Annu. Rev. Neurosci.*, vol. 19, pp. 577–621, Mar. 1996.

[7] S. E. Palmer, "Hierarchical structure in perceptual representation," *Cognit. Psychol.*, vol. 9, no. 4, pp. 441–474, 1977.

[8] M. W. O. E. Wachsmuth and D. I. Perrett, "Recognition of objects and their component parts: Responses of single units in the temporal cortex of the macaque," *Cerebral Cortex*, vol. 4, no. 5, pp. 509–522, 1994.

[9] M. Heiler and C. Schnörr, "Larning sparse representations by non-negative matrix factorization and sequential cone programming," *J. Mach. Learn. Res.*, vol. 7, pp. 1385–1407, Jul. 2006.

[10] P. O. Hoyer, "Non-negative sparse coding," in *Proc. 12th IEEE Workshop Neural Netw. Signal Process.*, Mar. 2002, pp. 557–565.

[11] P. O. Hoyer, "Non-negative matrix factorization with sparseness constraints," *J. Mach. Learn. Res.*, vol. 5, pp. 1457–1469, Dec. 2004.

[12] K. Yu, T. Zhang, and Y. Gong, "Nonlinear learning using local coordinate coding," in *Proc. 23rd Annu. Conf. Neural Inf. Process. Syst.*, 2009, pp. 1–9.

[13] J. Wang, J. Yang, K. Yu, F. Lv, T. S. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. Conf. Comput. Vis. Pattern Recognit.*, 2010, pp. 3360–3367.

[14] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and Y. Ma, "Robust face recognition via sparse representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 210–227, Feb. 2009.

[15] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Proc. Adv. Neural Inf. Process. Syst.*, 2001, pp. 556–562.

[16] P. Pentti and T. Unto, "Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values," *Environmetrics*, vol. 5, no. 2, pp. 111–126, 1994.

[17] C.-J. Lin, "On the convergence of multiplicative update algorithms for nonnegative matrix factorization," *IEEE Trans. Neural Netw.*, vol. 18, no. 6, pp. 1589–1596, Nov. 2007.

[18] D. Cai, X. He, X. Wu, and J. Han, "Non-negative matrix factorization on manifold," in *Proc. Int. Conf. Data Mining*, 2008, pp. 1–10.
[19] H. Kim and H. Park, "Sparse non-negative matrix factorizations via alternating non-negativity-constrained least squares for microarray data analysis," *Bioinformatics*, vol. 23, no. 12, pp. 1495–1502, 2007.
[20] D. Cai, X. He, X. Wang, H. Bao, and J. Han, "Locality preserving nonnegative matrix factorization," in *Proc. 21st Int. Joint Conf. Artif. Intell.*, 2009, pp. 1010–1015.
[21] X. Li and Y. Pang, "Deterministic column-based matrix decomposition," *IEEE Trans. Knowl. Data Eng.*, vol. 22, no. 1, pp. 145–149, Jan. 2010.
[22] Y. Yuan, X. Li, Y. Pang, X. Lu, and D. Tao, "Binary sparse nonnegative matrix factorization," *IEEE Trans. Circuits Syst. for Video Technol.*, vol. 19, no. 5, pp. 772–777, May 2009.
[23] D. Cai, X. He, J. Han, and T. S. Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 8, pp. 1548–1560, Aug. 2011.
[24] D. Cai, X. He, and J. Han, "Locally consistent concept factorization for document clustering," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 6, pp. 902–913, Jun. 2011.
[25] V. Monga and M. K. Mihçak, "Robust and secure image hashing via non-negative matrix factorizations," *IEEE Trans. Inf. Forensics Security*, vol. 2, no. 3, pp. 376–390, Sep. 2007.
[26] Y. Xu, Z. Zhang, P. Yu, and B. Long, "Pattern change discovery between high dimensional data sets," in *Proc. 20th ACM Int. Conf. Inf. Knowl. Manage.*, 2011, pp. 1097–1106.
[27] T. Hofmann, "Probabilistic latent semantic indexing," in *Proc. 22nd Annu. Int. ACM SIGIR Conf. Res. Develop. Inf. Retr.*, 1999, pp. 50–57.
[28] Y. Pang, Q. Hao, Y. Yuan, T. Hu, R. Cai, and L. Zhang, "Summarizing tourist destinations by mining user-generated travelogues and photos," *Comput. Vis. Image Understand.*, vol. 115, no. 3, pp. 352–363, 2011.
[29] J. Kivinen and M. K. Warmuth, "Additive versus exponentiated gradient updates for linear prediction," in *Proc. 27th Annu. ACM Symp. Theory Comput.*, 1995, pp. 209–218.
[30] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge, MA: MIT Press, 2001.
[31] G. W. Stewart, *Matrix Algorithms*, vol. 1. Philadelphia, PA: SIAM, 1998.
[32] D. Cai, X. He, and J. Han, "Using graph model for face analysis," Dept. Comput. Sci., UIUC, Urbana, Tech. Rep. UIUCDCS-R-2005-2636, Sep. 2005.
[33] D. Cai, X. Wang, and X. He, "Probabilistic dyadic data analysis with local and global consistency," in *Proc. 26th Annu. Int. Conf. Mach. Learn.*, 2009, pp. 105–112.
[34] S. Deerwester, S. T. Dumais, G. W. Furnas, T. K. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *J. Amer. Soc. Inf. Sci.*, vol. 41, no. 6, pp. 391–407, 1990.
[35] A. Y. Ng, M. I. Jordan, and Y. Weiss, "On spectral clustering: Analysis and an algorithm," in *Advances in Neural Information Processing Systems*. Cambridge, MA: MIT Press, 2001, pp. 849–856.
[36] D. Cai, X. He, and J. Han, "Document clustering using locality preserving indexing," *IEEE Trans. Knowl. Data Eng.*, vol. 17, no. 12, pp. 1624–1637, Dec. 2005.
[37] L. Lovasz and M. Plummer, *Matching Theory*. New York: Akadémiai Kiadó, 1986.
[38] B. A. Olshausen and D. J. Field, "Emergence of simple-cell receptive field properties by learning a sparse code for natural images," *Nature*, vol. 381, no. 6583, pp. 607–609, 1996.
[39] B. A. Olshausen and D. J. Field, "Sparse coding with an overcomplete basis set: A strategy employed by V1?" *Vis. Res.*, vol. 37, no. 23, pp. 3311–3325, 1997.
[40] J. F. Murray and K. Kreutz-Delgado, "Learning sparse overcomplete codes for images," *J. VLSI Signal Process. Syst.*, vol. 46, no. 1, pp. 97–110, 2006.
[41] J. Mairal, F. Bach, J. Ponce, and G. Sapiro, "Online learning for matrix factorization and sparse coding," *J. Mach. Learn. Res.*, vol. 11, pp. 19–60, Dec. 2010.
[42] A. P. Dempster, N. M. Laird, and D. B. Rubin, "Maximum likelihood from incomplete data via the em algorithm," *J. Royal Stat. Soc., Ser. B*, vol. 39, no. 1, pp. 1–38, 1977.

Yan Chen received the Bachelor's degree in mathematics from Jiangnan University, Wuxi, China, and the Master's degree in computer science from Zhejiang University, Hangzhou, China, in 2009 and 2012, respectively.

He is currently with the State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University. His current research interests include machine learning and computer vision.

Jiemi Zhang received the B.S. degree in mathematics from Southeast University, Nanjing, China. She is currently pursuing the Master's degree with the State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China.

Her current research interests include machine learning, computer vision, and multimedia information retrieval.

Deng Cai (M'09) received the Ph.D. degree in computer science from the University of Illinois at Urbana-Champaign, Urbana, in 2009.

He is currently an Associate Professor with the State Key Laboratory of CAD&CG, College of Computer Science, Zhejiang University, Hangzhou, China. His current research interests include machine learning, data mining, computer vision, and information retrieval.

Wei Liu received the M.Phil. and Ph.D. degrees in electrical engineering from Columbia University, New York, NY, in 2012.

He was an Intern with the Kodak Research Laboratories and IBM Thomas J. Watson Research Center, in 2010 and 2011, respectively. He is currently the Josef Raviv Memorial Post-Doctoral Fellow with the IBM Thomas J. Watson Research Center, Yorktown Heights, NY. His current research interests include machine learning, computer vision, pattern recognition, and information retrieval.

Dr. Liu was a recipient of the 2011-2012 Facebook Fellowship.

Xiaofei He (SM'10) received the B.S. degree from Zhejiang University, Hangzhou, China, and the Ph.D. degree from the University of Chicago, Chicago, IL, in 2000 and 2005, respectively, both in computer science.

He is currently a Professor with the State Key Laboratory of CAD&CG, Zhejiang University. He was a Research Scientist with Yahoo! Research Labs, Burbank, CA. His current research interests include machine learning, information retrieval, and computer vision.