PARALLEL VECTOR FIELD REGULARIZED NON-NEGATIVE MATRIX FACTORIZATION FOR IMAGE REPRESENTATION

Yong Peng*, Rixin Tang, Wanzeng Kong, Feiwei Qin

Feiping Nie

School of Computer Science and Technology Hangzhou Dianzi University Hangzhou 310018, China Center for OPTIMAL Northwestern Polytechnical University Xi'an 710072, China

ABSTRACT

Non-negative Matrix Factorization (NMF) is a popular model in machine learning, which can learn parts-based representation by seeking for two non-negative matrices whose product can best approximate the original matrix. However, the manifold structure is not considered by NMF and many of the existing work use the graph Laplacian to ensure the smoothness of the learned representation coefficients on the data manifold. Further, beyond smoothness, it is suggested by recent theoretical work that we should ensure second order smoothness for the NMF mapping, which measures the linearity of the NMF mapping along the data manifold. Based on the equivalence between the gradient field of a linear function and a parallel vector field, we propose to find the NMF mapping which minimizes the approximation error, and simultaneously requires its gradient field to be as parallel as possible. The continuous objective function on the manifold can be discretized and optimized under the general NMF framework. Extensive experimental results suggest that the proposed parallel field regularized NMF provides a better data representation and achieves higher accuracy in image clustering.

Index Terms— Non-negative matrix factorization, Vector field, Image representation, Clustering

1. INTRODUCTION

Matrix factorization is a class of effective methods to learn data representation. Among existing matrix factorization methods such as LU-decomposition, QR-decomposition, Cholesky decomposition, and Singular Value Decomposition (SVD), the Non-negative Matrix Factorization (NMF) is an very popular one which can learn parts of objects like human faces and text documents [1, 2]. Though the parts-based representation have got physiological evidence from previous studies [3] and shown great performance in face recognition [4] and document clustering [5], NMF fails to consider the geometrical structure of the data space which has been proven to be essential for data clustering and classification. To compensate the limitation, Cai et al. proposed to encode the geometrical information of the data space by constructing a nearest neighbor graph and formulated the Graph regularized Non-negative Matrix Factorization (GNMF) model [6, 7], which shows superior performance to NMF. However, such graph Laplacianbased method only enforces the smoothness of data representation on data manifold in first order manner. Based on the conclusions that 1) the second smoothness measures the linearity of the mapping function, 2) the gradient field of a linear function has to be a parallel vector field, which were shown in [8, 9], we propose a novel matrix factorization algorithm, called Parallel Field Regularized NMF (PFNMF), which enforces the learned codes to vary linearly along the data manifold. Specifically, we firstly estimate parallel vector fields on manifold from data samples, and then learn the codes by requiring the gradient fields of NMF mapping are as close to the estimated parallel fields as possible.

The remainder of this paper is organized as follows. Section 2 introduces the formulation of the proposed PFNMF model and its optimization. Experimental setting and results are presented in Section 3. Section 4 concludes the paper.

2. PARALLEL FIELD REGULARIZED NMF

2.1. NMF

NMF is a matrix factorization algorithm that focuses on analysis of data matrices whose elements are non-negative. Given a data matrix $X = [x_1, \dots, x_n] \in \mathbb{R}^{m \times n}$, NMF aims at finding two non-negative matrices $U = [u_{ik}] \in \mathbb{R}^{m \times K}$ and $H = [h_{jk}] \in \mathbb{R}^{n \times K}$, whose product can approximate X. Based on the square of the Euclidean distance between two matrices, the objective function of NMF can be formulated as

$$\mathcal{O}_{\rm NMF} = \|X - UH^T\|^2. \tag{1}$$

The K columns of U are basis vectors, and each column of H is an encoding of a sample vector in X and is an one-to-one mapping. The non-negative property enforced on both U and

^{*}Corresponding author(yongpeng@hdu.edu.cn). This work was supported by NSFC (61602140,61502129), Zhejiang Science and Technology Program (2017C33049,LQ16F020004), China Postdoctoral Science Foundation (2017M620470), Jiangsu Key Lab. of Image and Video Understanding for Social Safety (30916014107), Jiangsu Key Lab. of Big Data Security & Intelligent Processing (BDSIP1804) and Co-Innovation Center for Information Supply & Assurance Technology, Anhui University (ADXXBZ201704).

H allows only addictive combinations among different bases, which makes NMF learn parts-based representation.

2.2. Parallel Fields and Linear Functions

In geometry and vector calculus, a vector field is a mapping from a manifold \mathcal{M} to tangent spaces [10]. As shown in [8], the parallel vector field has close relationship to a linear function on the manifold, which can be described below.

Definition 1. (*Parallel Field* [10]). A vector field X on manifold \mathcal{M} is a parallel field if $\nabla X \equiv 0$, where ∇ is the covariant derivative on \mathcal{M} .

Definition 2. (*Linear Function* [10]). A continuous function $f : \mathcal{M} \to \mathbb{R}$ is said to be linear if $(f \circ \gamma)(t) = f(\gamma(0)) + ct$ for each geodesic γ .

In this paper, a function f is linear means that it varies linearly along the geodesics of the manifold. This definition is a natural extension of linear functions on Euclidean space. The following proposition reveals the relationship between a parallel vector field and a linear function on the data manifold. We will not strictly distinguish between the concepts of covariant derivative and gradient field in this paper.

Proposition 1. [10] Let V be a parallel field on the manifold. If it is also a gradient field for function $f, V = \nabla f$, then f is a linear function on the manifold.

2.3. PFNMF Model Formulation

As discussed before, we aim to ensure the linearity of NM-F embedding with respect to the data manifold, which is equivalent to the parallelism of the gradient field of the NMF mapping. However, it is difficult to design constraints on the gradient field of a mapping function directly [8]. Therefore, we propose to 1) learn vector fields on manifold from data samples to approximate the gradient field of the NMF mapping function, and encourage the vector fields to be as parallel as possible; 2) learn NMF embedding while enforcing the encodes to be as close to the estimated parallel fields as possible.

Since the manifold \mathcal{M} is unknown, the mapping function in NMF $f_k(x_j) = h_{jk} \triangleq f_j^{(k)}, j = 1, \ldots, n$ has no explicit form. Following [8], we introduce how to estimate the tangent space of each data point, which is important for discretizing the continuous objective function form when estimating parallel vector fields and learning NMF embedding.

Let W be the corresponding affinity matrix of graph \mathcal{G} and W is simply defined by 0-1 weight. Then, for each x_i , we can estimate its tangent space $T_{x_i}\mathcal{M}$ by performing P-CA on its local neighborhood. We choose the eigenvectors corresponding to the d largest eigenvalues since $T_{x_i}\mathcal{M}$ is ddimensional. Let $T_i \in \mathbb{R}^{m \times d}$ be the matrix whose columns constitute an orthogonal basis for $T_{x_i}\mathcal{M}$. It is easy to show that $P_i = T_i T_i^T$ is the *unique* orthogonal projection from \mathbb{R}^m onto the tangent space $T_{x_i}\mathcal{M}$. That is, for any vector $a \in \mathbb{R}^m$, we have $P_i a \in T_{x_i}\mathcal{M}$ and $(a - P_i a) \perp P_i a$. Estimating the Parallel Vector Field. Firstly, we try to find vector fields which are as parallel as possible on the manifold. Let V be a smooth vector field on \mathcal{M} . By definition, the covariant derivative of V should be zero. That is, $\nabla V \equiv 0$. Naturally, the objective for estimating parallel vector field is

$$E(V) = \int_{\mathcal{M}} \|\nabla V\|_{\mathrm{HS}}^2 dx, \qquad (2)$$

where $\|\cdot\|_{HS}$ denotes Hilbert-Schmidt tensor norm [11].

For each point x_i , let V_{x_i} denote the value of the vector field V at x_i , and $\nabla V|_{x_i}$ denote the value of ∇V at x_i . According to its definition, V_{x_i} should be a vector in the tangent space $T_{x_i}\mathcal{M}$. Therefore, it can be represented by the local coordinates of the tangent space, $V_{x_i} = T_i v_i$, where $v_i \in \mathbb{R}^d$.

By discretizing (2), the parallel field V can be obtained by solving the following optimization problem (Here we use the notation \mathbb{V} instead of V while the specific definition of \mathbb{V} will be given next.):

$$\min_{\mathbb{V}} E(\mathbb{V}) = \sum_{i=1}^{n} \|\nabla V|_{x_i}\|_{\mathrm{HS}}^2.$$
 (3)

Then according to the analysis in [8], the optimization problem of (3) reduces to

$$\min_{\mathbb{V}} E(\mathbb{V}) = \sum_{i,j=1}^{n} w_{ij} \|P_i T_j v_j - T_i v_i\|^2.$$
(4)

The compact form of $E(\mathbb{V})$ is $E(\mathbb{V}) \triangleq \mathbb{V}^T B\mathbb{V}$ if we define $Q_{ij} = T_i^T T_j, \mathbb{V} = [v_1^T, v_2^T, \cdots, v_n^T]^T \in \mathbb{R}^{dn}$ and

$$B = \begin{pmatrix} B_{11} \cdots B_{1n} \\ \vdots \ddots \vdots \\ B_{n1} \cdots B_{nn} \end{pmatrix},$$
(5)

where B is a $dn \times dn$ sparse block matrix. If we index each $d \times d$ block by B_{ij} , then for j = 1, 2, ..., n, we have

$$B_{jj} = \sum_{x_i \in N_k(x_j)} w_{ij} (Q_{ij} Q_{ij}^T + I_d), \tag{6}$$

$$B_{ij} = \begin{cases} -2w_{ij}Q_{ij}, & \text{if } x_i \sim x_j \\ 0, & \text{otherwise.} \end{cases}$$
(7)

Obviously, \mathbb{V} is a *dn*-dimensional big column vector concatenating all the v_i 's.

We impose a global normalization constraint $\|V\|^2 = 1$ on vector field; thus, the Lagrangian function w.r.t. E(V) is

$$\mathcal{L}_{E(\mathbb{V})} = E(\mathbb{V}) - \lambda \left(\mathbb{V}^T \mathbb{V} - 1 \right).$$
(8)

The parallel vector field \mathbb{V} can be estimated by solving the following eigenvalue decomposition problem

$$B\mathbb{V} = \lambda \mathbb{V}.\tag{9}$$

The multiple vector fields are eigenvectors corresponding to the smallest eigenvalues of the matrix B. Specifically, the dimensionality of learned NMF embedding is K; thus, we use $\tilde{\mathbb{V}} \in \mathbb{R}^{dn \times K}$ to denote the stack of K eigenvectors.

Learning NMF Embedding. Once the parallel vector fields V_i are obtained, the embedding functions $f^{(k)} : \mathcal{M} \to \mathbb{R}$ can be constructed by requiring their gradient fields to be as close as V_i as possible. This can be attained in the least square sense and thus minimizing the following objective

$$R(f^{(k)}) = \int_{\mathcal{M}} \|\nabla f^{(k)} - V\|^2 dx$$
 (10)

Based on the analysis in [8], $R(f^{(k)})$ can be discretized as $R(f^{(k)}) = \sum_{i,j=1}^{n} w_{ij}((x_j - x_i)^2 T_i v_i - f_j^{(k)} + f_i^{(k)})^2.$ (11)

By incorporating this as a regularizer into the objective of NMF, we obtain the objective of PFNMF as

$$\mathcal{O}_{\text{PFNMF}} = \|X - UH^T\|^2 + \alpha \sum_k R(f^{(k)}) \quad (12)$$

be the graph Laplacian. Then we have

Let L be the graph Laplacian. Then we have

$$R(f^{(k)}) = 2f^{(k)T}Lf^{(k)} + \sum_{i,j=1}^{n} w_{ij} \left((x_j - x_i)^2 T_i v_i \right)^2$$

$$-2\sum_{i,j=1}^{n} w_{ij} (x_j - x_i)^T T_i v_i s_{ij}^T f^{(k)},$$
where $s_{ij} \in \mathbb{P}^n$ is a selection vector of all zero element

where $s_{ij} \in \mathbb{R}^n$ is a selection vector of all zero elements except for the *i*-th element being -1 and the *j*-th element being 1. We further construct a $dn \times dn$ block diagonal matrix G, and a $dn \times n$ block matrix $C = [C_1^T, C_2^T, \dots, C_n^T]^T$. Let G_{ii} denote the *i*-th $d \times d$ diagonal block of G, and C_i denote the *i*-th $d \times n$ block of C, we define:

$$G_{ii} = \sum_{j \sim i}^{n} w_{ij} T_i^T (x_j - x_i) (x_j - x_i)^T T_i$$

$$C_i = \sum_{j \sim i}^{n} w_{ij} T_i^T (x_j - x_i) s_{ij}^T$$
(13)

With simple algebra transformations, it is easy to check that $R(f^{(k)})$ can be rewritten as follows:

$$R(f^{(k)}) = 2f^{(k)T}Lf^{(k)} + \mathbb{V}^T G \mathbb{V} - 2\mathbb{V}^T C f^{(k)}.$$
 (14)

Similar to the basic NMF, we need to minimize \mathcal{O}_{PFNMF} by updating one variable while fixing the other.

Let ψ_{ik} and ϕ_{jk} be the Lagrange multipliers for constraints $u_{ik} \ge 0$ and $h_{jk} \ge 0$. We define matrix $\Psi = [\psi_{ik}]$ and $\Phi = [\phi_{jk}]$, then the Lagrangian function $\mathcal{L}_{\text{PFNMF}}$ is

$$\mathcal{L}_{\text{PFNMF}} = \mathcal{O}_{\text{PFNMF}} + \operatorname{Tr}\left(\Psi U^{T}\right) + \operatorname{Tr}\left(\Phi H^{T}\right) \tag{15}$$

Fixing *H*, the partial derivative of $\mathcal{L}_{\text{PFNMF}}$ w.r.t. *U* is:

$$\frac{\partial \mathcal{L}_{\text{PFNMF}}}{\partial U} = -2\left(X - UH^T\right)H + \Psi.$$

Using the KKT condition $\psi_{ik}u_{ik} = 0$, we can get $-(XH)_{ik}u_{ik} + (UH^TH)_{ik}u_{ik} = 0$,

which leads to the following updating rule for u_{ik} $u_{ik} \leftarrow u_{ik}(XH)_{ik}/(UH^TH)_{ik}$.

 $u_{ik} \leftarrow u_{ik} (XH)_{ik} / (UH^T H)_{ik}.$ (16) Based on the NMF definition, the mapping function $F = [f^{(1)}, f^{(2)}, \cdots, f^{(K)}] \in \mathbb{R}^{n \times K}$ is defined as $f^{(k)} = h_{\cdot k} \in \mathbb{R}^{n \times 1}, f_j^{(k)} = h_{jk}$ and thus we have

$$\frac{\partial \mathcal{L}_{\text{PFNMF}}}{\partial H} = -2\left(X - UH^T\right)U + \alpha \frac{\partial R}{\partial F} + \Phi$$

 $= -2X^TU + 2HU^TU + 4\alpha LH - 2\alpha C^T \tilde{\mathbb{V}} + \Phi$ Using KKT conditions $\phi_{jk}h_{jk} = 0$, we have

$$-(X^T U)_{jk}h_{jk} + (HU^T U)_{jk}h_{jk} + 2\alpha(LH)_{jk}h_{jk}$$

$$\alpha(C^T \mathbb{V})_{jk} h_{jk} = 0$$

Since NMF only allows additive combinations, we decompose L and \mathbb{V} as L = D - W and $\tilde{\mathbb{V}} = \tilde{\mathbb{V}}^+ - \tilde{\mathbb{V}}^-$, $\tilde{\mathbb{V}}^+ = (|\tilde{\mathbb{V}}| + \tilde{\mathbb{V}})/2$, $\tilde{\mathbb{V}}^- = (|\tilde{\mathbb{V}}| - \tilde{\mathbb{V}})/2$, then we obtain the following updating rule for h_{jk} :

$$h_{jk} \leftarrow h_{jk} \frac{(X^T U + \alpha C^T \tilde{\mathbb{V}}^+ + 2\alpha W H)_{jk}}{(H U^T U + \alpha C^T \tilde{\mathbb{V}}^- + 2\alpha D H)_{jk}}$$
(17)

We can easily find that PFNMF will boil down to NMF when α =0 and GNMF [7] if we do not consider the vector fields $\tilde{\mathbb{V}}$.

The workflow of PFNMF is summarized in Algorithm 1.

Algorithm 1 PFNMF

Input: Data samples $X = (x_1, x_2, ..., x_n) \in \mathbb{R}^{m \times n}$ and α ; **Output:** The basis matrix U and coefficient matrix H. **for** i = 1 to n **do** Compute tangent spaces $T_{x_i}\mathcal{M}$ for each data sample by performing PCA on neighborhood of x_i ; **end for** Construct matrix B according to (7); Do eigen-decomposition on (9) to estimate $\tilde{\mathbb{V}}$; // Iteratively optimize PFNMF model **while** not converged **do** Update U according to (16) with H fixed; Update H according to (17) with U fixed;

end while

3. EXPERIMENTS

We investigate the effectiveness of PFNMF on image clustering. We set the parameter K to be the number of clusters and use the obtained coefficient matrix H to determine the cluster label of each data point. α is searched from $2^{-10:10}$.

3.1. Data Sets and Experimental Settings

Data Sets. Three data sets are used in this experiment. The important statistics of these data sets are summarized below (see also Table 1). **COIL20**. It contains 20 objects. The images of each object were taken 5 degrees apart as the object is rotated on a turntable and each object has 72 images. The size of each image is 32×32 pixels, with 256 gray levels per pixel. Thus, each image is represented by a 1024-dimensional vector. **ORL**. 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. **CMU PIE**. It contains 32×32 gray scale images of 68 subjects. Each person has 42 facial images under different light and illumination conditions.

Evaluation Metrics. The clustering result is evaluated by comparing the obtained label of each sample with the label provided by the data set. Two standard clustering metrics, the accuracy (ACC) and normalized mutual information (NMI), are used to measure the clustering performance.

 Table 1. Properties of the used data sets.

dataset	#size	#dimensionality	#class
COIL20	1440	1024	20
ORL	400	4096	40
PIE	2856	1024	68

3.2. Clustering Results

To show the improvement of the clustering performance by our method, we compared PFNMF with the following three popular algorithms: K-means, NMF [1], and GNMF [7]. The evaluations were conducted with different numbers of clusters. For each given cluster numbers, 20 test runs were conducted on different randomly chosen clusters. The final performance is recorded by averaging the performance of the 20 tests. For fair comparison, we record the randomly selected cluster indices, and fix them for all competing algorithms.

Table 2. Clustering performance on COIL20.

K	Accuracy (mean±std-dev%)						
	Kmeans	NMF	GNMF	PFNMF			
4	84.11±12.15	79.77±14.26	92.73±11.98	95.97±8.74			
6	78.07±12.16	75.53 ± 11.87	86.66±12.67	92.82±8.20			
8	72.96±10.23	72.06 ± 9.64	90.43±7.75	95.29±6.48			
10	68.87±6.17	$68.89 {\pm} 8.88$	81.41±7.58	87.44±6.51			
12	68.20 ± 4.02	67.87 ± 5.45	79.13±5.96	86.17±5.32			
14	67.11±5.65	$66.67 {\pm} 4.60$	$82.49 {\pm} 4.61$	84.67±4.09			
16	65.23 ± 4.42	65.56 ± 4.69	79.09 ± 4.10	80.90±3.72			
18	62.74±3.85	63.15 ± 3.65	78.97±3.49	80.41±4.29			
20	60.49	58.33	80.69	85.14			
Avg.	69.75	68.65	83.51	87.65			
K	Normalized Mutual Information (mean±std-dev%)						
	Kmeans	NMF	nformation (mean±s GNMF	PFNMF			
4	75.03±16.69	70.75±17.67	90.32±13.14	94.55±10.96			
6	74.83±12.79	71.99±11.91	87.91±8.84	92.54±7.75			
8	74.23±7.65	72.02 ± 7.57	91.29 ± 5.72	95.36±5.23			
10	72.68 ± 5.95	71.72 ± 7.49	$86.80{\pm}5.07$	90.41±4.68			
12	73.22 ± 3.25	72.36 ± 3.95	86.83±3.43	90.85±3.41			
14	74.19±3.93	72.92 ± 3.75	89.29 ± 2.80	91.33±2.51			
16	73.88 ± 2.71	72.61 ± 3.38	88.36 ± 2.10	89.73±2.58			
18	73.25 ± 2.54	$72.28 {\pm} 2.41$	88.36 ± 1.57	89.50±1.61			
20	73.86	71.51	89.12	90.50			
Avg.	73.91	72.02	88.70	91.64			

Table 3. Clustering performance on ORL.

			01					
	K	Accuracy (mean±std-dev%)						
	11	Kmeans	NMF	GNMF	PFNMF			
	2	90.25±14.46	89.25±12.38	93.00±12.71	93.25±11.62			
	4	80.37±11.51	68.75±9.58	81.38±11.11	85.00±8.51			
	8	66.68 ± 8.74	67.06±7.18	70.56 ± 7.28	72.63±10.14			
	12	$62.58 {\pm} 6.58$	63.50±6.81	68.42 ± 7.26	69.67±6.51			
	16	59.94 ± 5.91	59.53±4.67	64.63±6.11	67.69±5.18			
	20	57.30 ± 4.92	54.75±4.99	62.47 ± 4.05	65.45±5.65			
	25	$56.18 {\pm} 4.04$	53.98±4.29	61.38±3.25	65.10±4.15			
ĺ	30	55.63 ± 3.33	52.23±3.29	59.33±2.91	61.83±3.34			
	40	53.50	49.75	59.75	62.25			
	Avg.	64.71	62.09	68.99	71.46			
	V	Normalized Mutual Information (mean±std-dev%)						
	K	Kmeans	NMF	GNMF	PFNMF			
	2	71.86±37.30	65.61±36.27	78.85±35.24	78.87±32.65			
ĺ	4	75.17±13.04	69.20±11.23	74.17±14.14	77.62±11.41			
	8	70.97 ± 7.31	69.86±6.91	72.94 ± 7.25	75.37±7.78			
	12	$71.94{\pm}5.16$	71.79±5.28	75.50 ± 4.63	76.99±5.13			
	16	70.45 ± 4.68	70.01±3.80	73.48 ± 4.62	76.04±3.93			
	20	70.59 ± 3.86	68.55±3.17	74.76 ± 3.01	76.99±3.54			
	25	71.13 ± 2.88	69.50±2.82	74.62 ± 2.16	77.50±2.79			
	30	71.33 ± 2.05	69.98±2.65	74.20 ± 1.99	76.01±2.11			
	40	71.42	68.92	75.70	78.23			
	Avg.	71.65	68.09	74.91	77.07			

Table 2, 3 and 4 show the clustering results on the data sets COIL20, ORL and PIE. The best results are shown in boldface. From these tables, we can find PFNMF consistently

Table 4. Clustering performance on PIE.

K	Accuracy (mean±std-dev%)						
n	Kmeans	NMF	GNMF	PFNMF			
10	31.15 ± 4.00	56.62 ± 5.16	85.01±9.28	91.39±7.38			
20	27.20 ± 1.99	$55.32 {\pm} 4.46$	$80.69 {\pm} 6.20$	83.72±5.64			
30	26.35 ± 1.29	$55.87 {\pm} 2.18$	$81.60 {\pm} 3.08$	84.19±3.51			
40	$25.34{\pm}1.31$	55.96 ± 3.37	77.51 ± 3.84	79.81±3.18			
50	$24.62{\pm}1.08$	$55.27 {\pm} 2.10$	$76.59 {\pm} 3.90$	78.83±4.41			
60	24.17 ± 1.10	$55.64{\pm}2.42$	$74.69 {\pm} 2.89$	76.34±3.29			
68	24.54	56.79	70.52	73.01			
Avg.	26.20	55.92	78.09	81.04			
K	Normalized Mutual Information (mean±std-dev%)						
Λ	Kmeans	NMF	GNMF	PFNMF			
10	37.75 ± 6.40	66.22 ± 2.88	89.53±5.28	91.16±4.23			
20	$44.24{\pm}2.30$	$72.77 {\pm} 2.88$	$88.85 {\pm} 2.86$	90.73±3.10			
30	$48.19 {\pm} 2.05$	$76.47 {\pm} 1.04$	$89.85 {\pm} 1.25$	91.12±1.81			
40	$50.08 {\pm} 1.68$	$78.20{\pm}1.34$	$88.86{\pm}1.53$	90.87±1.57			
50	$51.33 {\pm} 1.40$	$78.75 {\pm} 0.85$	88.67 ± 1.46	90.54±1.35			
60	52.67 ± 1.21	$80.36 {\pm} 0.99$	$88.36 {\pm} 0.92$	90.18±1.17			
68	53.77	80.18	87.21	89.96			
Avg.	48.29	76.14	88.76	90.65			

outperforms its counterpart GNMF by around 2%-3%, which means that the data representation learned by PFNMF show more respect to the underlying data manifold. The linearity of image data representation emphasized by the second order smoothness is beneficial to clustering task.

3.3. Basis Vectors and Image Encodings

In this test, we randomly select 25 subjects from the ORL data set and for each subject we randomly select 5 face images. Figure 1 shows the basis vectors respectively learned by NMF and PFNMF. Comparing the basis images obtained by PFNMF with the original face images. The basis learned by PFNMF is obviously much clear than that learned by N-MF, which will naturally lead to better face image encodings for clustering.

	際				00	98		
Te -	202			10 3		16	64	ie.
35 7 90		10.0	10			and a	25	
2012	11.3						19	20
a y y	1		10				法	F

(a) Original images.

(b) Basis by NMF. (c) Basis by PFNMF.

Fig. 1. Basis vectors learned from the ORL data set.. Large values are illustrated with white pixels.

4. CONCLUSIONS

We have presented a novel method for matrix factorization algorithm to enforce the second order smoothness of data representation called PFNMF. Experiments shows that our proposed method performs better than other comparison methods in image clustering. In the future, we will extend our method to deal with data in noisy environments and thus develop its robust version.

5. REFERENCES

- Daniel D Lee and H Sebastian Seung, "Learning the parts of objects by non-negative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–791, 1999.
- [2] Daniel D Lee and H Sebastian Seung, "Algorithms for non-negative matrix factorization," in Advances in Neural Information Processing Systems, 2000, pp. 556–562.
- [3] E Wachsmuth, MW Oram, and DI 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.
- [4] Stan Z Li, Xin Wen Hou, Hong Jiang Zhang, and Qian Sheng Cheng, "Learning spatially localized, partsbased representation," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2001, vol. 1, pp. 207– 212.
- [5] Wei Xu, Xin Liu, and Yihong Gong, "Document clustering based on non-negative matrix factorization," in *Proceedings of the 26th ACM SIGIR Conference on Research and Development in Information Retrieval*, 2003, pp. 267–273.
- [6] Deng Cai, Xiaofei He, Xiaoyun Wu, and Jiawei Han, "Non-negative matrix factorization on manifold," in *IEEE International Conference on Data Mining*, 2008, pp. 63–72.
- [7] Deng Cai, Xiaofei He, Jiawei Han, and Thomas S Huang, "Graph regularized nonnegative matrix factorization for data representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 33, no. 8, pp. 1548–1560, 2011.
- [8] Binbin Lin, Chiyuan Zhang, and Xiaofei He, "Semisupervised regression via parallel field regularization.," in Advances in Neural Information Processing Systems, 2011, vol. 24, pp. 433–441.
- [9] Binbin Lin, Xiaofei He, Chiyuan Zhang, and Ming Ji, "Parallel vector field embedding," *The Journal of Machine Learning Research*, vol. 14, no. 1, pp. 2945–2977, 2013.
- [10] Luther Pfahler Eisenhart, *Riemannian geometry*, Princeton university press, 1997.
- [11] Andreas Defant and Klaus Floret, "Tensor norms and operator ideals," 1993.