A METHOD OF INITIALIZATION FOR NONNEGATIVE MATRIX FACTORIZATION

Yong-Deok Kim and Seungjin Choi

Department of Computer Science, POSTECH, Korea
{*karma13, seungjin*}*@postech.ac.kr*

ABSTRACT

Nonnegative matrix factorization (NMF) is a widely-used method for multivariate nonnegative data analysis, due to its ability to learn a parts-based representation. However, the standard NMF algorithm does not always find spatially localized basis images in practice, unless sparseness constraints are employed. In this paper we present a method of structured initialization which enables the standard NMF algorithm to find spatially localized basis images. The initialization method is based on the hierarchical clustering of attributes through a similarity measure reflecting 'closeness to rank-one'. Numerical experiments with face image data, confirm the validity of our initialization method.

Index Terms— Feature extraction, matrix decomposition, pattern clustering methods, pattern classification, unsupervised learning

1. INTRODUCTION

Nonnegative matrix factorization (NMF) is a popular method for multivariate analysis of nonnegative data such as images, documents, and spectrograms [1, 2, 3, 4]. Successful applications of NMF result from its ability to learn a parts-based representation through a matrix factorization, $\boldsymbol{X} = \boldsymbol{AS}$, where $\boldsymbol{X} \in \mathbb{R}^{m \times N}$ is a data matrix which consists of N m-dimensional multivariate nonnegative data vectors, $\boldsymbol{A} \in \mathbb{R}^{m \times n}$ is a basis matrix, and $\boldsymbol{S} \in \mathbb{R}^{n \times N}$ is an encoding variable matrix. In contrast to principal component analysis (PCA) or independent component analysis (ICA) which finds a holistic representation, NMF was claimed to determine basis images that consist of distinguishable parts of face [1]. This characteristics popularizes NMF in pattern recognition and computer vision applications (for example, face recognition). However, in practice, NMF often fails to find a parts-based representation [5, 6], as shown in Fig. 1.

Fig. 1. Exemplary 20 basis images learned by NMF, from ORL face images, are shown. These basis images are closer to holistic representation, rather than parts-based representation, which is not a desirable characteristics of NMF.

Recently, various methods for improving NMF have been developed, most of which involve adding constraints such as locality or sparseness to the standard NMF error function [7, 5]. An exemplary modification is the local NMF (LNMF) [5] where it was shown that incorporating locality, orthogonality, and sparseness constraints, improved NMF such that better spatially localized basis images could be determined. However local NMF algorithms incorporating additive constraints require much more iterations to achieve the convergence, compared to the standard NMF algorithm.[1] Our empirical experience shows that LNMF takes at least 3 times more iterations to achieve the convergence, compared to the standard NMF algorithm.

In this paper we propose an alternative way for improving NMF to find localized basis images. Our approach is to initialize the standard NMF algorithm in such a way to find spatially localized basis images, rather than incorporating extra constraints. We show that our initialization method enables existing NMF algorithms to:

- find more localized basis images, compared to random initialization;
- speed up the convergence of existing NMF algorithms;
- find LMNF-like basis images very quickly, compared to LNMF.

The main idea of our initialization is to group attributes (pixels in images) into parts through the hierarchical clustering with a similarity measure reflecting 'closeness to rank-one'. Earlier work on structured initialization for NMF was reported in [8], where the spherical k-means clustering was employed to group data points (column vectors of $\boldsymbol{X}$), in order to improve NMF. In contrast, our initialization is to group attributes (row vectors of $\boldsymbol{X}$), placing an emphasis on a parts-based representation.

2. THE INITIALIZATION METHOD

2.1. Grouping

Our initialization method is motivated by a common sense on 'part', which is a smallest unit that has some perceptual meaning. For example, a face image consists of various parts, including eyes, nose, eyebrows, cheek, lip, and so on. Metaphorically, a pixel corresponds to an atom and then, a part can be considered as a molecule. As atoms in a molecule perform a chemical reaction together, pixels that build a part should be grouped together. The main idea of our initialization method is to group pixels that show the similar activity patterns, by investigating pixel variations across the time.

In order to illustrate this idea, we consider an ideal case. Suppose that the data matrix $\boldsymbol{X}$ is given by

$$[X_{it}] = \begin{bmatrix} 1 & 0 & 0 & 2 & 3 & 0 \\ 2 & 0 & 0 & 4 & 6 & 0 \\ 0 & 1 & 1 & 2 & 4 & 2 \\ 3 & 0 & 0 & 6 & 9 & 0 \\ 1 & 0 & 0 & 3 & 4 & 0 \end{bmatrix}, \quad (1)$$

where each row vector represents variations of a pixel across the time. In this example, each image consists of 5 pixels and we have

[1]The standard NMF algorithm means Lee-Seung's NMF algorithm in their paper [1, 2].

6 images in total. One can easily see that pixels 1, 2, and 4 show the same activity pattern, i.e., row vectors 1, 2, and 4 are linearly dependent. This suggests a grouping of row vectors 1, 2, and 4. On the other hand, pixels 1 and 3 can't be grouped together, since their activity patterns are different. With the same reason, pixels 1 and 5 are not grouped together. It follows from these intuitive observations that we conclude:

- Pixels 1, 2, and 4 are grouped together as a part.
- The row vectors $\boldsymbol{X}_{1,:}, \boldsymbol{X}_{2,:}, \boldsymbol{X}_{4,:}$ are parallel to each other, where $\boldsymbol{X}_{i,:}$ represents the ith row vector in the matrix $\boldsymbol{X}$.
- rank $\left(\boldsymbol{X}_{(1,2,4),:}\right) = 1$, where $\boldsymbol{X}_{(1,2,4),:}$ is a sub-matrix consists of the row vectors 1, 2, and 4 of $\boldsymbol{X}$.

Following the observations in this simple example, we introduce a 'closeness to rank-one' (CRO) measure in order to investigate whether row vectors in the sub-matrix show similar patterns or not. The CRO measure is defined by

$$\text{CRO}\left(\boldsymbol{X}_{(i,j,\ldots,k),:}\right) \quad = \quad \frac{\sigma_1^2}{\sum_{i=1}^{r}\sigma_i^2} = \frac{\sigma_1^2}{||\boldsymbol{X}_{(i,j,\ldots,k),:}||_F^2},$$

where $\sigma_1 \geq \sigma_2 \geq \cdots \geq \sigma_r \geq 0$ are singular values of the sub-matrix $\boldsymbol{X}_{(i,j,\ldots,k),:}$ and r is the rank of $\boldsymbol{X}_{(i,j,\ldots,k),:}$.

Before initializing the basis matrix $\boldsymbol{A}$, we first group pixel profiles (corresponding to row vectors of $\boldsymbol{X}$) through the hierarchical clustering where CRO in (2) is used as the similarity measure. The outline of the CRO-based hierarchical clustering algorithm is summarized in Table 1.

Table 1. Algorithm outline: CRO-based hierarchical clustering.

Input: $\boldsymbol{X} \in \mathbb{R}_+^{m\times N}$.
Output: n clusters.
Step 1 (Initialization)
· Assign each row vector $\boldsymbol{X}_{1,:}, \ldots, \boldsymbol{X}_{m,:}$ into each own clusters $C_1, \ldots, C_m$.
· Calculate CRO in (2) between every pairs of clusters.
Repeat Steps 2 and 3 until n clusters remains.
Step 2 (Merging)
· Find a pair of clusters with the largest CRO.
· Merge them into a single cluster.
Step 3 (Updating CRO)
· Compute CRO between the newly-merged cluster and remaining clusters.

In the case of the data matrix $\boldsymbol{X}$ given in (1), the CRO-based hierarchical clustering algorithm provides a dendrogram shown in Fig. 2. In such a case, taking 3 clusters, i.e., $n = 3$, yields that row vectors 1, 2, and 4 constitute a cluster and row vector 5 and 3 are associated with each individual cluster, respectively. Following this clustering result, the basis matrix $\boldsymbol{A}$ is initialized as

$$\boldsymbol{A} \quad = \quad \begin{bmatrix} 0.27 & \epsilon & \epsilon \\ 0.53 & \epsilon & \epsilon \\ \epsilon & \epsilon & 1.00 \\ 0.80 & \epsilon & \epsilon \\ \epsilon & 1.00 & \epsilon \end{bmatrix}, \qquad (2)$$

where column vectors are orthogonal each other, reflecting 3 distinctive parts, ϵ is a small positive constant and the vector $[0.27\ 0.53\ 0.80]^\top$ is the largest left singular vector of $\boldsymbol{X}_{(1,2,4),:}$. Some small positive value $\epsilon > 0$ is used instead of 0 to prevent parameters from being locked under NMF multiplicative update rules. More detailed explanation, including the justification for using the left singular vector $[0.27\ 0.53\ 0.80]^\top$, is given in Sec. 2.3.

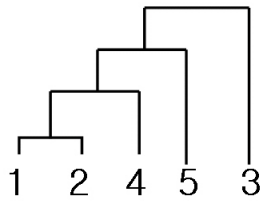


Fig. 2. The dendrogram resulting from the CRO-based hierarchical clustering, is shown in the case of the simple example where the data matrix is given in (1).

2.2. Implementation

The CRO-based hierarchical clustering method heavily relies on singular value decomposition SVD, requiring the calculation of the largest singular vector of a matrix which reflects two clusters in consideration. In other words, every time the CRO between two clusters is updated in Step 3 (in Table 1), an execution of SVD is required, which demands high computational complexity. Here we present an efficient implementation of the CRO-based hierarchical clustering method, where we introduce a method of merging two rank-one approximations.

The sub-matrices considered in Step 3, are 'fat' matrices where the number of columns are much greater than the number of rows. Suppose that a sub-matrix $\boldsymbol{X} \in \mathbb{R}^{r\times N}$ ($r \ll N$) is given. The largest singular vector of this $\boldsymbol{X}$ can be calculated through applying SVD to $\boldsymbol{X}\boldsymbol{X}^\top \in \mathbb{R}^{r\times r}$

$$\begin{aligned} \boldsymbol{X}\boldsymbol{X}^\top &= \left(\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\right)\left(\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top\right) \\ &= \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top\boldsymbol{V}\boldsymbol{\Sigma}^\top\boldsymbol{U}^\top \\ &= \boldsymbol{U}\boldsymbol{\Sigma}^2\boldsymbol{U}^\top, \end{aligned} \qquad (3)$$

where the SVD of $\boldsymbol{X}$ is $\boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. The $\boldsymbol{V}$ is obtained by $\boldsymbol{V}^\top = \boldsymbol{\Sigma}^{-1}\boldsymbol{U}^\top\boldsymbol{X}$. The size of $\boldsymbol{X}\boldsymbol{X}^\top$ is much smaller than the one of $\boldsymbol{X}$, which dramatically reduce the computational load in SVD. In fact, this trick is known as the *snap-shot* method.

The Step 1 in Table 1, requires the SVD of $2 \times N$ matrices in order to calculate the CRO for every pairs of clusters. The snap-shot method is applied to calculate the largest singular vector of each sub-matrix.

The most serious bottleneck is in Step 3, where we need to calculate the largest singular vector of sub-matrices associated with every possible pairs of clusters. Instead of direct calculation of SVD of the sub-matrix, we develop a method of merging two eigen-space models, each of which is represented by rank-one approximation. Suppose that $\boldsymbol{X}_1 \in \mathbb{R}^{m_1\times N}$ and $\boldsymbol{X}_2 \in \mathbb{R}^{m_2\times N}$ are the matrices, each of which is associated with a cluster that is determined in the hierarchical clustering. The rank-one approximations of these matrices are given by $\boldsymbol{X}_1 \approx \sigma_1\boldsymbol{u}_1\boldsymbol{v}_1^\top$ and $\boldsymbol{X}_2 \approx \sigma_2\boldsymbol{u}_2\boldsymbol{v}_2^\top$, where $\boldsymbol{u}_i$ and $\boldsymbol{v}_i$ are left and right singular vectors associated with the largest

singular value of $\boldsymbol{X}_i$ for $i = 1, 2$. The concatenated matrix $\boldsymbol{X}$ has the following decomposition:

$$\begin{aligned} \boldsymbol{X} &= \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \end{bmatrix} \approx \begin{bmatrix} \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top \\ \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^\top \end{bmatrix} \\ &= \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{u}_2 \end{bmatrix} \begin{bmatrix} \sigma_1 \boldsymbol{v}_1^\top \\ \sigma_2 \boldsymbol{v}_2^\top \end{bmatrix} \\ &= \boldsymbol{L}\boldsymbol{R}. \end{aligned} \tag{4}$$

The snap-shot method is applied to $\boldsymbol{R} \in \mathbb{R}^{2\times N}$, in order to determine the decomposition $\boldsymbol{R} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^\top$. Then we have $\boldsymbol{X} \approx \widetilde{\boldsymbol{U}}\boldsymbol{\Sigma}\boldsymbol{V}^\top$, where $\widetilde{\boldsymbol{U}} = \boldsymbol{L}\boldsymbol{U}$. Then the rank-one approximation of $\boldsymbol{X}$ is determined by $\boldsymbol{X} \approx \sigma \widetilde{\boldsymbol{u}} \boldsymbol{v}^\top$, where $\boldsymbol{u}$ and $\boldsymbol{v}$ are left and right singular vectors associated with the largest singular value σ of $\boldsymbol{R}$, and $\widetilde{\boldsymbol{u}} = \boldsymbol{L}\boldsymbol{u}$.

Note that $\boldsymbol{X} \approx \sigma \widetilde{\boldsymbol{u}} \boldsymbol{v}^\top$ is not the exact rank-one truncated SVD of $\boldsymbol{X}$, since the rank-one approximation is applied to $\boldsymbol{L}\boldsymbol{R}$, not directly to $\boldsymbol{X}$. But approximation error is not big, so the accumulation of errors never yields that a pair of clusters which is merged, is not the one that is associated with the highest CRO measure.

2.3. Initialization

We illustrate how the basis matrix $\boldsymbol{A} \in \mathbb{R}^{m\times n}$ and encoding variable matrix $\boldsymbol{S} \in \mathbb{R}^{n\times N}$ are initialized, using the CRO-based hierarchical clustering result. Suppose that given the nonnegative data matrix $\boldsymbol{X} \in \mathbb{R}^{m\times N}$, the CRO-based hierarchical clustering produces n clusters, $\mathcal{C}_1, \ldots, \mathcal{C}_n$ and their associated sub-matrix, $\boldsymbol{X}_1, \ldots, \boldsymbol{X}_n$, where the associated sub-matrix $\boldsymbol{X}_p = \boldsymbol{X}_{(i_p, j_p, \ldots, k_p),:}$ contains the row vectors of $\boldsymbol{X}$ that are grouped as the cluster $\mathcal{C}_p$. Each column vector of $\boldsymbol{A}$ and row vector of $\boldsymbol{S}$ are initialized according to the rank-one approximation of the associated sub-matrix. There exists a permutation matrix $\boldsymbol{P}$ such that $\boldsymbol{P}\boldsymbol{X} = \left[\boldsymbol{X}_1^\top, \ldots, \boldsymbol{X}_n^\top\right]^\top$. The rank-one approximation of each sub-matrix $\boldsymbol{X}_p$, leads to

$$\boldsymbol{P}\boldsymbol{X} = \begin{bmatrix} \boldsymbol{X}_1 \\ \boldsymbol{X}_2 \\ \vdots \\ \boldsymbol{X}_p \\ \vdots \\ \boldsymbol{X}_n \end{bmatrix} \approx \begin{bmatrix} \sigma_1 \boldsymbol{u}_1 \boldsymbol{v}_1^\top \\ \sigma_2 \boldsymbol{u}_2 \boldsymbol{v}_2^\top \\ \vdots \\ \sigma_p \boldsymbol{u}_p \boldsymbol{v}_p^\top \\ \vdots \\ \sigma_n \boldsymbol{u}_n \boldsymbol{v}_n^\top \end{bmatrix} \tag{5}$$

$$= \begin{bmatrix} \boldsymbol{u}_1 & \boldsymbol{0} & \ldots & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{u}_2 & \ldots & \boldsymbol{0} & \ldots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots & & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{u}_p & \ldots & \boldsymbol{0} \\ \vdots & \vdots & & \vdots & & \vdots \\ \boldsymbol{0} & \boldsymbol{0} & \ldots & \boldsymbol{0} & \ldots & \boldsymbol{u}_n \end{bmatrix} \begin{bmatrix} \sigma_1 \boldsymbol{v}_1^\top \\ \sigma_2 \boldsymbol{v}_2^\top \\ \vdots \\ \sigma_p \boldsymbol{v}_p^\top \\ \vdots \\ \sigma_n \boldsymbol{v}_n^\top \end{bmatrix} \tag{6}$$

which is $(\boldsymbol{P}\boldsymbol{A})\boldsymbol{S}$ for initialization. Replacing zeros by ϵ's, the pth column vector of $\boldsymbol{A}$ and the pth row vector of $\boldsymbol{S}$ are initialized as:

$$\boldsymbol{A}_{(i_p, j_p, \ldots, k_p), p} = \boldsymbol{u}_p, \tag{7}$$

$$\boldsymbol{A}_{\setminus(i_p, j_p, \ldots, k_p), p} = \epsilon, \tag{8}$$

$$\boldsymbol{S}_{p,:} = \sigma_p \boldsymbol{v}_p^\top, \tag{9}$$

where $\setminus(i_p, j_p, \ldots, k_p)$ represents the rest of elements excluding $(i_p, j_p, \ldots, k_p)$ in the pth column vector of $\boldsymbol{A}$. The initialized matrix $\boldsymbol{A}$ in (2) is an instance of (7) and (8). In our experiment, we used $\epsilon \in [0.0005, 0.05]$. The choice of ϵ, indeed, have influences on the final result of the standard NMF. These influences are examined in next section.

3. NUMERICAL EXPERIMENTS

We show the useful behavior of our initialization method, with two face image datasets, comparing it to random initialization method. Face image datasets used in numerical experiments are: (1) MIT CBCL dataset [9]; (2) AT&T and Cambridge University, ORL dataset [10]. In the case of CBCL dataset, the size of each face image is 19×19 and for ORL data set, each face image is resized by 28×23.

We compare our initialization method to random initialization, investigating how goodness-of-fit (GOF) and sparseness changes after the convergence of the standard NMF algorithm starting from two different initialization methods. In experiments, we used LS NMF algorithm, the updating rules of which are given in [2]. The GOF used in experiments, is nothing but the relative reconstruction error defined by GOF $= \|\boldsymbol{X} - \boldsymbol{A}\boldsymbol{S}\|_F / \|\boldsymbol{X}\|_F$.

In regards to parts-based representation, we evaluate the sparseness that was used in [7], defined by

$$\text{sparseness}(\boldsymbol{a}) = \frac{1}{\sqrt{m}-1}\left(\sqrt{m} - \frac{\|\boldsymbol{a}\|_1}{\|\boldsymbol{a}\|_2}\right), \quad \boldsymbol{a} \in \mathbb{R}^m. \tag{10}$$

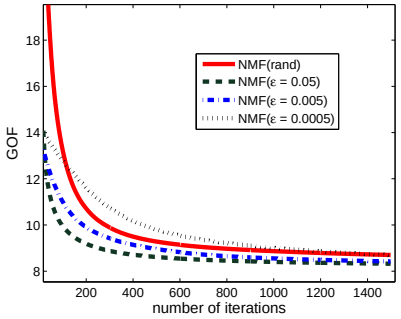
The sparseness measure (10) evaluates to 1 if and only if $\boldsymbol{a}$ contains only single nonzero component, and takes a value of 0 if and only if all components are equal, interpolating smoothly between these two extremes.

The initialization for the basis matrix $\boldsymbol{A}$, involves ϵ-padding as well as singular vectors regarding the CRO-based hierarchical clustering. Certainly, values of ϵ have influences on GOF and sparseness. In experiments, we used several different values of $\epsilon \in \{0.05, 0.01, 0.005, 0.001, 0.0005\}$. We also investigated the performance, for different values of n (the number of basis vectors), $n \in \{25, 36, 49, 64\}$. Evolutions of GOF and sparseness are shown in Fig. 3, where experiments were carried out using random initialization and the proposed initialization method with different choices of ϵ. In these experiments, CBCL dataset was used, with $n = 49$. In case of random initialization, the best performance was chosen among 10 independent trials. We observed that

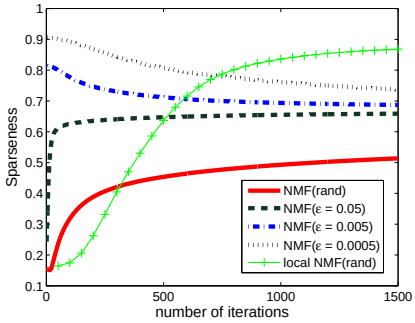
1. Our initialization method always produced more sparse representation, compared to random initialization.
2. As increasing ϵ from 0.0005 to 0.05, we lose sparseness while gaining GOF.
3. Taking large ϵ yields slightly better goodness-of-fit, more sparse representation, and faster convergence, compared to random initialization. See Fig. 3 and Fig. 4.
4. Taking small ϵ with early stopping, yields LNMF-like result in terms of goodness-of-fit as well as parts-based representation. See Fig. 3 and Fig. 5.

4. CONCLUSIONS

We have presented a method of initialization for existing NMF algorithms, leading them to find more spatially localized parts-based representation, compared to random initialization method. The core idea of the initialization was to group attributes into clusters, each of which might correspond to a part. The grouping was carried out through the hierarchical clustering where we have introduced a similarity measure, CRO, which reflects redundancy among vectors in



(a)



(b)

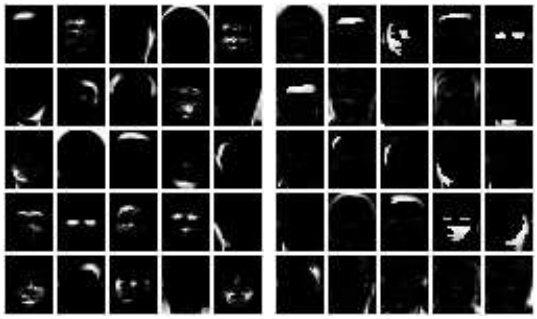
Fig. 3. Evolutions of: (a) GOF; (b) sparseness, are shown in cases of various values of ϵ in the proposed initialization, compared to random initialization, with CBCL dataset.

(a) (b)

Fig. 4. Basis images of CBCL data are shown, determined by two initialization methods: (a) random initialization; (b) the proposed initialization method with $\epsilon = 0.05$.

terms of linear dependency. An efficient implementation was presented, where the CRO between clusters was calculated by merging two rank-one approximations. Numerical experiments have confirmed that the initialization method indeed found spatially localized parts-based representations.

Acknowledgments: This work was supported by Korea MCIE under Brain Neuroinformatics Program, and Korea MIC under ITRC support program supervised by the IITA (IITA-2006-C1090-0603-0045).



(a) (b)

Fig. 5. Basis images of ORL data are shown, determined by (a) LNMF (random initialization and 1500 iterations), GOF=20.67%, sparseness=79.28%; (b) NMF (our initialization method with $\epsilon = 0.0005$ and 120 iterations), GOF=20.54%, sparseness=76.89%.

5. REFERENCES

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

[2] D. D. Lee and H. S. Seung, "Algorithms for non-negative matrix factorization," in *Advances in Neural Information Processing Systems*. 2001, vol. 13, MIT Press.

[3] Y. -C. Cho and S. Choi, "Nonnegative features of spectro-temporal sounds for classfication," *Pattern Recognition Letters*, vol. 26, no. 9, pp. 1327–1336, 2005.

[4] M. Kim and S. Choi, "Monaural music source separation: Nonnegativity, sparseness, and shift-invariance," in *Proc. Int'l Conf. Independent Component Analysis and Blind Signal Separation*, Charleston, South Carolina, 2006, pp. 617–624, Springer.

[5] S. Z. Li, X. W. Hou, H. J. Zhang, and Q. S. Cheng, "Learning spatially localized parts-based representation," in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, Kauai, Hawaii, 2001, pp. 207–212.

[6] D. L. Donoho and V. Stodden, "When does nonnegative matrix facotrization give a correct decomposition into parts?," in *Advances in Neural Information Processing Systems*. 2004, vol. 16, MIT Press.

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

[8] S. Wild, J. Curry, and A. Dougherty, "Improving non-negative matrix factorizations through structured initialization," *Pattern Recognition*, vol. 37, pp. 2217–2232, 2004.

[9] B. Weyrauch, J. Huang, B. Heisele, and V. Blanz, "Component-based face recognition with 3D morphable models," in *Proc. First IEEE Workshop on Face Processing in Video*, Washington, D.C., 2004.

[10] F. Samaria and A. Harter, "Parameterisation of a stochastic model for human face identification," in *Proc. 2nd IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, 1994.