Binary Sparse Nonnegative Matrix Factorization

Yuan Yuan, *Member, IEEE*, Xuelong Li, *Senior Member, IEEE*, Yanwei Pang, *Member, IEEE*, Xin Lu, and Dacheng Tao, *Member, IEEE*

***Abstract*— This paper presents a fast part-based subspace selection algorithm, termed the *binary sparse nonnegative matrix factorization* (B-SNMF). Both the training process and the testing process of B-SNMF are much faster than those of *binary principal component analysis* (B-PCA). Besides, B-SNMF is more robust to occlusions in images. Experimental results on face images demonstrate the effectiveness and the efficiency of the proposed B-SNMF.**

***Index Terms*— Fast algorithms, non-negative matrix factorization, part-based representation, sparseness, subspace selection.**

I. Introduction

SUBSPACE SELECTION is a process of choosing a subspace from high dimensional space and projecting samples onto the selected subspace. It not only reduces the computational cost of data processing but also improves, in most cases, the classification accuracy.

Subspace selection algorithms have been widely utilized in face recognition, image retrieval, data de-noising, and human gait recognition. Representative algorithms include principal component analysis (PCA) [16], Fisher's linear discriminant analysis (LDA) [12], general averaged divergence analysis [22], discriminative locality alignment [3], [9], 2-D subspace methods [1], [2], geometric mean based discriminative subspace selection [10], Laplacianfaces [5], simplexization [8], general tensor discriminant analysis (GTDA) [11].

Recently, endeavor has been done to speed up both the training process and the projection step [4], [6], [7]. As an enhanced version of LDA, spectral regression discriminant analysis (SRDA) [4], [7] reduces the computational cost of its training process by casting the linear discriminant analysis into a regression framework with the help of spectral graph analysis [9]. Instead of focusing on the speed of training process, binary principal component analysis (B-PCA) [17] aims to speed up the projection process. To see the advantages of B-PCA, let us investigate the computational cost of projecting a vectorized testing image $\mathbf{x} \in R^{D\times 1}$ onto a principal vector $\mathbf{u} \in R^{D\times 1}$ of PCA, where $D = w \times h$ equals to the number of image pixels and w is the image width and h the height. The projection process $\mathbf{u}^T\mathbf{x}$ needs D floating point multiplications and $D-1$ floating point additions. The time T for projecting the image onto N base vectors (i.e. principal vectors) of PCA is

$$T = N \times w \times h \times T_{fm} + N \times (w \times h - 1) \times T_{fa}$$

where T_{fm} is the time occupied by a single floating point multiplication, while T_{fa} is the time occupied by a single floating point addition [17]. When the size of an image is large, the size of an image database is large or computational resource is limited, the computational cost of projection bottlenecks the real-time applications.

BPCA [17] alleviates the computational cost problem of PCA. As a fast implementation of PCA, B-PCA employs Haar-like box functions to linearly represent each basis vector that spans PCA subspace. Because the inner product of a Haar-like box function and an image can be computed very efficiently with the help of the integral image [20], the process of projecting the image onto the approximated basis vectors is very fast. The computational cost is determined by the number of Haar-like functions and is independent of the image size.

However, the training stage of B-PCA is too long to be accepted, especially for large size images. It takes a great deal of time to select a small number of Haar-like box functions, because the Haar-like box functions have to be selected from a huge dictionary and each selection process involves the matrix inverse operation which is time consuming. The Haar-like box functions can be divided into one-box functions and symmetric two-box functions (as shown in Fig. 1). Suppose the image size is $w \times h$, then there are $h \times (h-1) \times w \times (w-1)/4$ one-box functions and $h \times (h+1) \times w \times (w-1)/16$ symmetric two-box functions in the dictionary. In summary, B-PCA suffers from the disadvantages of long training time, which impedes it from real-time applications.

In this paper, we propose a fast part-based subspace selection method, which is termed the binary sparse non-negative matrix factorization (B-SNMF). It consists of two main advantages in comparing with B-PCA. First, both the training speed and the testing speed of B-SNMF are much higher than those of B-PCA. Second, B-SNMF is occlusion-insensitive for recognition tasks while the performance of B-PCA degenerates drastically when occlusion occurs. Specifically, we propose to select the Haar-like box functions by a

Manuscript received May 1, 2008; revised August 16, 2008. First version published March 16, 2009; current version published June 10, 2009. This work was supported in part by the National Natural Science Foundation of China (No. 60605005), the Open Fund of National Laboratory of Pattern Recognition, China, and the Nanyang Technological University Start-Up Grant (No. M58020010). This paper was recommended by Associate Editor D. Schonfeld.

Y. Yuan is with the School of Engineering and Applied Science, Aston University, Birmingham B4 7ET, U.K.

X. Li is with the School of Computer Science and Information Systems, Birkbeck College, University of London, London WC1 E7HX, U.K.

Y. Pang and X. Lu are with the School of Electronic Information Engineering, Tianjin University, Tianjin 300072, China.

D. Tao is with School of Computer Engineering, Nanyang Technological University, Singapore, 639798 (e-mail: dctao@ntu.edu.sg).

Digital Object Identifier 10.1109/TCSVT.2009.2017306

1051-8215/$25.00 © 2009 IEEE

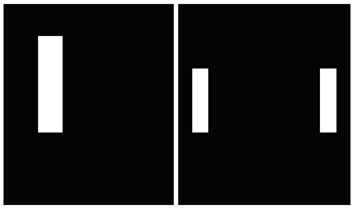


Fig. 1. Examples of Haar-like box functions: an one-box Haar-like function (left) and symmetric two-box Haar-like box function (right).

computationally economical "pre-filtering" step followed the conventional matching pursuit process. Because the sparseness of the basis vectors, fewer Haar-like box functions are required to obtain satisfying recognition accuracies and reconstruction error.

The rest of this paper is organized as follows: Section II gives an introduction to NMF and SNMF (sparse NMF). Section III presents the proposed B-SNMF. In Section IV experiments based on the face recognition are reported. Section V concludes the paper.

II. Brief Review of NMF and SNMF

B-SNMF incorporates merits from the non-negative matrix factorization (NMF) [15] and the sparse non-negative matrix factorization (SNMF) [14]. It can be deemed as the fast version of SNMF.

A. NMF

NMF is a part-based algorithm to represent an image (e.g face image) as a linear combination of basis vectors [15]. Its basis vectors are not orthogonal but non-negative to mimic the representation process in human brain according to evidences from psychological and physiological researches. NMF allows additive combination only for basis images to represent an image. With this restriction, locality property has been shown in both basis and coefficients from the intuitive views of the image parts.

Assume each image is vectorized to a vector in a high-dimensional space R^D where D equals to the image size $w \times h$. Given the non-negative training images (vectors) $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_N]$, the NMF finds a $D \times d$ non-negative transformation matrix $\mathbf{U} = [\mathbf{u}_1 \cdots \mathbf{u}_d]$, where the columns (basis vectors) of $\mathbf{U}$ span a subspace in a low-dimensional subspace R^d, and the non-negative matrix $\mathbf{Y} = \left[\mathbf{y}_1 \cdots \mathbf{y}_N\right] \in R^{d\times N}$ such that $\mathbf{X} \approx \mathbf{UY}$. Lee and Sung [15] proposed to find optimal matrices $\mathbf{U}$ and $\mathbf{Y}$ by maximizing the following objective function

$$\begin{aligned}&\min_{\mathbf{U},\mathbf{Y}} \sum_{i=1}^{N}\sum_{j=1}^{D}\left[x_{ij}\log(\mathbf{UY})_{ij} - (\mathbf{UY})_{ij}\right]\\ &\text{s.t.}\quad U_{ij} \geq 0, Y_{ij} \geq 0\end{aligned} \tag{1}$$

where x_{ij} and $(\mathbf{UY})_{ij}$ are the ijth entries of matrices $\mathbf{X}$ and $\mathbf{UY}$, respectively. This objective function is related to the likelihood of generating the images in $\mathbf{X}$ from the basis $\mathbf{U}$ and encoding $\mathbf{Y}$ [15].

B. SNMF

Originally, NMF is assumed to be able to find part-based, linear representations of a set of non-negative samples. However, recent research on NMF has discovered that NMF does not always result in part-based representations. To guarantee part-based characteristics, Hoyer [14] explicitly imposed sparseness constraints on the squared error based objective function. This sparse constrained method is termed SNMF in this paper. It exhibits different levels of part-based (locality) property by specifying different value of sparseness. The sparseness is measured by

$$sparseness\,(\mathbf{x}) = \frac{\sqrt{D} - \left(\sum |x_i|\right)/\sqrt{\sum x_i^2}}{\sqrt{D} - 1} \tag{2}$$

where x_i is the i element of vector $\mathbf{x}$ and D is the dimensionality of $\mathbf{x}$. The function value is one if and only if $\mathbf{x}$ contains only a single non-zero component, and takes a value of zero if and only if all components are identical [14].

The problem of SNMF can be formulated as [14]

$$\begin{aligned}&\min_{W,H} \|\mathbf{X} - \mathbf{UY}\|^2\\ &\text{s.t.}\quad \mathbf{U}, \mathbf{Y} \geq 0\\ &\qquad sparseness(\mathbf{u}_i) = S_u,\ \forall i\\ &\qquad sparseness(\mathbf{y}_i) = S_y\ ,\ \forall j.\end{aligned} \tag{3}$$

If the sparseness is forced to be close to 1, the subspace analysis becomes holistic-based and SNMF will degenerate to NMF.

III. B-SNMF

A. Training Process

Binary SNMF (B-SNMF) is an approximate version of the original SNMF in the sense that the basis vector $\mathbf{u}$ of SNMF is approximated by linearly combining a small number K of selected Haar-like box functions $\mathbf{v}_i$

$$f(\mathbf{u}) \approx \tilde{\mathbf{u}} = \sum_{i=0}^{K-1} c_i \mathbf{v}_i \tag{4}$$

where $f(\mathbf{u})$ is a reshape operation to re-arrange $\mathbf{u}$ as a $w \times h$ (image size) matrix. The Haar-like box functions $\mathbf{v}$ can take two forms [17], [20]

$$v(i,j) = \begin{cases}1, & l \leq i \leq r \text{ and } t \leq j \leq b\\ 0, & \text{elsewhere}\end{cases}, \tag{5}$$

and

$$v(i,j) = \begin{cases}1, & l_1 \leq i \leq r_1 \text{ and } t_1 \leq j \leq b_1\\ 1, & l_2 \leq i \leq r_2 \text{ and } t_2 \leq j \leq b_2\ ,\\ 0, & \text{elsewhere}\end{cases} \tag{6}$$

where (5) corresponds to a one-box Haar-like box function and (6) is a symmetric two-box Haar-like box function. As stated in [17], the number of single and that of double Haar-like box functions are $h \times (h-1) \times w \times (w-1)/4$ and $h \times (h+1) \times w \times (w-1)/16$, respectively. The union Θ of the one-box and two-box Haar-like box functions is called the dictionary of the Haar-like box functions. The size D of the dictionary is given by $D = h\times(h-1)\times w\times(w-1)/4+h\times(h+1)\times w\times(w-1)/16$.

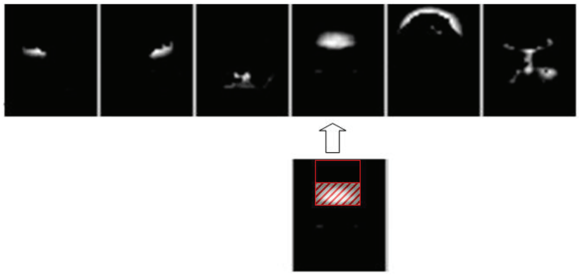


Fig. 2. Top: Example basis vectors of SNMF with sparseness≈0.8; Bottom: The basis vector of SNMF can be approximated very well by only one Haar function illustrated as the rectangle

The K Haar-like functions expressed as (5) and (6) are selected from the huge dictionary. To speed up the selection process, we propose a computationally economical "pre-filtering" step followed by a matching pursuit process [13], [17], [18]. The pre-filtering can exclude a great deal of the Haar-like box functions in the huge dictionary with little computational cost. Only those Haar-like box functions that pass the pre-filtering step are to be evaluated by the time consuming matching pursuit criterion. In B-PCA, there is no such pre-filtering step and so all the Haar-like box functions have to be evaluated by the time-consuming criterion. Thus, the training time of B-PCA is very long relative to that of B-SNMF. The pre-filtering is motivated by the sparseness property (as shown in Fig. 2) of SNMF.

The pre-filtering step can exclude the useless Haar-like box functions by simply counting the number of non-zero values of the part of the basis vector $f(\mathbf{u})$ overlapped by the Haar-like box $\mathbf{v} \in \Theta$. The percentage z of non-zero elements of the intersection between $f(\mathbf{u})$ and $\mathbf{v}$ is

$$z = \frac{\sum_{i=l}^{r}\sum_{j=t}^{t} I(f(\mathbf{u};i,j)) > 0)}{(r-l+1)\times(b-t+1)} \times 100 \tag{7}$$

where $f(\mathbf{u};i,j)$ is the ij entry of the function $f(\mathbf{u})$ and the value of indicate function $I(f(\mathbf{u};i,j) > 0) = 1$ if $f(\mathbf{u};i,j) > 0$, otherwise the value of the indicator $I(f(\mathbf{u};i,j) > 0) = 0$. If z is less than a predefined threshold Z (e.g., 0.01), the Haar-like box function $\mathbf{v}$ should not be used to approximate $\mathbf{u}$. Because $\mathbf{u}$ is usually sparse and compact, $z < Z$ occurs in many locations. Hence, many Haar-like box functions satisfying $z < Z$ can be excluded from the candidate. We call these Haar-like box functions easy-excluded Haar-like box functions. Denote the easy-excluded set of Haar-like box functions satisfying $z < Z$ by $\boldsymbol{\Omega}$. Only the Haar-like box functions in $\mathbf{v}_i \in \Theta - \boldsymbol{\Omega}$ are evaluated by the following time consuming matching pursuit process which involves the matrix inverse operation: suppose $t-1$ Haar-like functions $\mathbf{V}_{t-1} = \left[v_1 \ \cdots \ v_{t-1}\right]$ have been selected, the t Haar-like box function $\mathbf{v}_t$ is selected according to [13], [17], [18]

$$\mathbf{v}_t = \arg\max_{\mathbf{v}_i \in \Omega} \frac{\left[\mathbf{u} - \mathbf{V}_{t-1}\left(\mathbf{V}_{t-1}^T\mathbf{V}_{t-1}\right)^{-1}\mathbf{V}_{t-1}^T\mathbf{u}\right]^T \times \left[\mathbf{v}_i - \mathbf{V}_{t-1}\left(\mathbf{V}_{t-1}^T\mathbf{V}_{t-1}\right)^{-1}\mathbf{V}_{t-1}^T\mathbf{v}_i\right]}{||\mathbf{u} - \mathbf{V}_{t-1}\left(\mathbf{V}_{t-1}^T\mathbf{V}_{t-1}\right)^{-1}\mathbf{V}_{t-1}^T\mathbf{u}||}. \tag{8}$$

The numerator of (8) is a product of two items. The first item is the component of $\mathbf{u}$ that is orthogonal to the subspace spanned by the columns of $\mathbf{V}_{t-1}$. The second item is the reconstruction residual of $\mathbf{v}$ using the reconstruction matrix $\mathbf{V}_{t-1}\left(\mathbf{V}_{t-1}^T\mathbf{V}_{t-1}\right)^{-1}\mathbf{V}_{t-1}^T$. Under the special condition that $\mathbf{V}_{t-1}$ is an orthogonal matrix, the reconstruction matrix becomes $\mathbf{V}_{t-1}\mathbf{V}_{t-1}^T$.

In B-PCA, all Haar-like box functions in the huge dictionary are subject to the selection process in (8). So the selection process in B-PCA is very slow. The proposed method B-SNMF can alleviate the difficulty by filtering out the easy-excluded Haar- like box functions.

More Haar-like box functions should be selected until the angle α between $\mathbf{u}$ and $\tilde{\mathbf{u}}$

$$\alpha(\mathbf{u},\hat{\mathbf{u}}) = \frac{180}{\pi}\frac{\mathbf{u}^T\tilde{\mathbf{u}}}{(||\mathbf{u}||)\,(||\tilde{\mathbf{u}}||)} \tag{9}$$

is below a threshold θ.

In B-SNMF, SNMF is conducted as many times as the number of features to be extracted. At each time i, the input of SNMF is the reconstruction error $\mathbf{E}$ corresponding to the $i-1$ available basis vectors $\tilde{\mathbf{U}} = \left[\tilde{\mathbf{u}}_1 \ \cdots \tilde{\mathbf{u}}_{i-1}\right]$, i.e.,

$$\begin{gathered}\mathbf{E} = \mathbf{X} - \tilde{\mathbf{X}} \\ \tilde{\mathbf{X}} = \tilde{\mathbf{U}}\left(\tilde{\mathbf{U}}^T\tilde{\mathbf{U}}\right)^{-1}\tilde{\mathbf{U}}^T\mathbf{X}\end{gathered} \ . \tag{10}$$

In (10), $\mathbf{X}$ contains all training samples and the second row is the reconstruction formula for non-orthogonal basis vectors [21].

The output of SNMF is the most important basis vector and other basis vectors are discarded for further processing. At the beginning of the training process $\mathbf{E}$ is initialized to be $\mathbf{X}$.

The framework of B-SNMF is similar to that of B-PCA. But in comparing with B-PCA, both the training and testing processes of B-SNMF are much faster.

B. Feature Extraction Process

Denote $\mathbf{x} \in \mathbb{R}^{w\times h}$ as an image from which features are to be extracted. The feature extraction can be done by the inner product of the matrix $\mathbf{x}$ with $f(\mathbf{u})$

$$y = sum(f(\mathbf{u}) \otimes \mathbf{x}) \tag{11}$$

where $\otimes$ stands for the Hadamard product and the summation is $sum(\mathbf{x}) = \sum_{i=0}^{h-1}\sum_{j=0}^{w-1} x(i,j)$. Though (11) is straightforward, it is not efficient in computation. In fact, B-SNMF as well as B-PCA calculates y based on the integral image $it(\mathbf{x})$. The ij entry of $it(\mathbf{x})$ is

$$it(\mathbf{x};i,j) = \sum_{r=0}^{i}\sum_{c=0}^{j} x(r,c). \tag{12}$$

To see the usefulness of the integral image, let us substitute (4) into (11)

$$\begin{aligned} y &= sum(f(\mathbf{u}) \otimes \mathbf{x}) \\ &\approx sum\left(\sum_{i=0}^{K-1} c_i \mathbf{v}_i \otimes \mathbf{x}\right) \\ &= \sum_{i=0}^{K-1} c_i sum(\mathbf{v}_i \otimes \mathbf{x}). \end{aligned} \tag{13}$$

Without loss of generality, we assume $\mathbf{v}_i$ is a one-box Haar-like function. Because of

$$\begin{aligned} sum(\mathbf{v}_i \otimes \mathbf{x}) &= \sum_{r=t_i}^{b_i} \sum_{c=l_i}^{r_i} x(r, c) \\ &= it(\mathbf{x}; b, r) - it(\mathbf{x}; b, l-1) \\ &\quad - it(\mathbf{x}; t-1, r) + it(t-1, l-1) \end{aligned} \tag{14}$$

it holds that

$$\begin{aligned} y = \sum_{i=0}^{K-1} c_i \big[& it(\mathbf{x}; b, r) - it(\mathbf{x}; b, l-1) \\ & - it(\mathbf{x}; t-1, r) + it(t-1, l-1) \big]. \end{aligned} \tag{15}$$

It is worth noting that (15) is irrelevant to the image size. Therefore, if K is small, the right hand side of (15) can be computed fast. Equation (11) needs $w \times h$ multiplications and $w \times h - 1$ additions while (15) needs K multiplications and $4K - 1$ additions.

The testing process of B-SNMF is faster than that of B-PCA because the number K in (15) in B-SNMF is usually smaller that that in B-PCA. To clarify this point, it is helpful to observe the appearance of $f(\mathbf{u}_i)$ as shown in Fig. 2. Obviously, the basis vectors are sparse, compact and have good local property in comparing with that of PCA (as shown in Fig. 3). In Fig. 2, the black means zero value and the white indicates positive value. The characteristic of SNMF makes it possible to represent the basis vector using smaller number of Haar-like box functions to approximate the basis vector than that for approximating the basis vector of PCA. Fox example, in the bottom of Fig. 2 the basis vector can be approximated very well (the reconstruction error ratio is less than 0.1) by only one Haar-like box function illustrated as the Haar-like rectangles. The Haar-like box functions exhibit the unique characteristic of having both global and local properties and are more suitable for analyzing the signal in Fig. 2 than the basis vector (i.e., eigenface) as shown in Fig. 3. Fox example, about 100 Haar-like functions are necessary to approximate the basis vector of PCA as shown in the first image of Fig. 3 so that the reconstruction error ratio is not larger than 0.1.

IV. Experimental Results

In the following experiments, ORL database [19] is employed to evaluate the performance of the proposed B-SNMF. The ORL database is made up of forty subjects, with which each subject has ten images. In our experiments, five images of each subject have been randomly chosen for training and the remaining five for testing. Totally, five groups of training and testing sets are generated by this way.

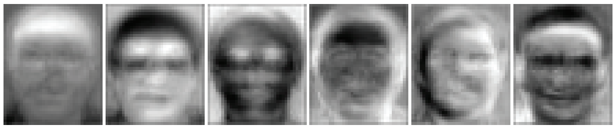


Fig. 3. Example basis vectors of PCA. Note the holistic characteristic of the basis vectors.

SNMF is equivalent to regular NMF plus sparseness constraints. In other words, regular NMF is a special case of SNMF. Let the sparseness coefficients of SNMF be 0, SNMF degenerates to regular NMF. In our experiments, we tuned the sparseness coefficients such that we can get parts-based representations and the recognition rate is the highest. Our experimental results show that: when the sparseness coefficients $S_u = 0.8$ and $S_y = 0.7$, both SNMF and our B-SNMF works very well. The angle [as shown in (9)] threshold of B-SNMF is $\theta = 25$.

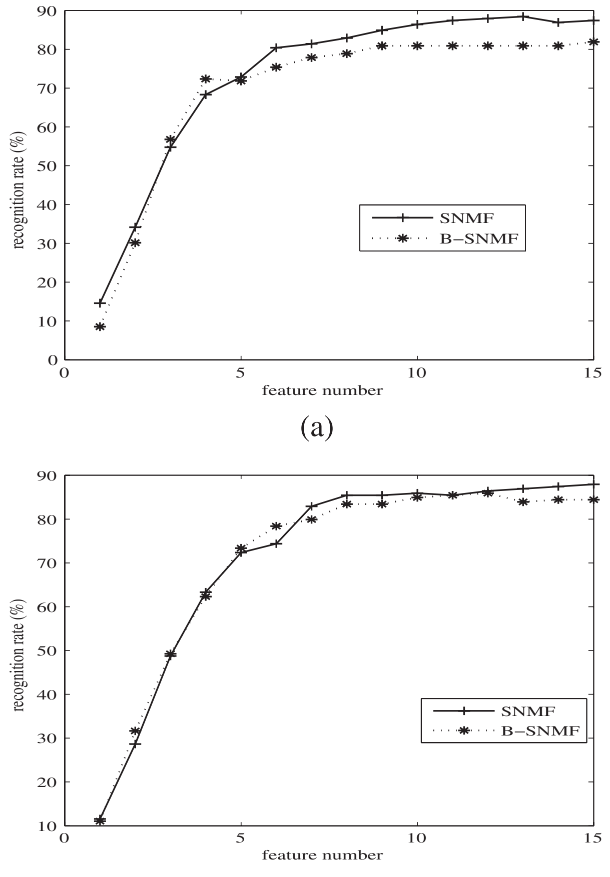
Fig. 4(a) shows the average recognition accuracies of B-SNMF and SNMF with the image size 14×12. The setup of the experiments corresponding to Fig. 4(b) is identical to that of Fig. 4(a) except that in Fig. 4(b) the image size is 44×36. As shown in both Fig. 4(a) and Fig. 4(b), the recognition rates of B-SNMF are as good as those of SNMF as long as the image size is large enough.

Fig. 5 compares the recognition performance of B-SNMF and SNMF with PCA when random occlusions (see Fig. 6) occur in testing images. Because we focus on unsupervised learning algorithms, comparison with supervised algorithms such as LDA and MFA is unfair. The image size is 44×36. For each testing image (there are 200 testing images), a white block is used to occlude the face image. The size of the white block varies from 8×8 to 14×14. The position of the white block is random. The sparseness and locality properties of B-SNMF make it much more robust to occlusions than both PCA and B-PCA. Because the recognition rates of B-PCA are not higher than those of PCA, they are lower than those of B-SNMF. In Fig. 5 as well as in Fig. 4(b), the recognition performance of B-SNMF is comparable to that of SNMF.

So far, we have demonstrated the performance of B-SNMF in terms of recognition accuracy. Let us now report its performance in aspect of computational cost.

Suppose the dictionary size is D, $|\Omega|$ is the cardinality of Ω and Ω is the set of Haar-like boxes satisfying $z < Z = 0.01$ (as shown in [7)]. Then the fraction, $|\Omega|/D$, of easy-excluded Haar-like box functions reflects the extent of reduced training cost of B-SNMF relative to B-PCA. The smaller the $|\Omega|/D$ is, the faster training speed of B-SNMF will be. Easy-excluded Haar-like box functions are free of being evaluated by (8). Table I shows the value of $|\Omega|/D$. The actual reduced training time can be derived from the value of $|\Omega|/D$. It is observed that the advantage of B-SNMF becomes remarkable with the increasing of the image size.

Table II compares the number of selected Haar-like box functions of B-SNMF and that of B-PCA. The number of Haar-like box functions increases until the reconstruction error



(a)

(b)

Fig. 4. Recognition accuracies for B-NMFSC and SNMF: (a) recognition rates for image size 14 × 12. (b) recognition rates for image size 44 × 36.

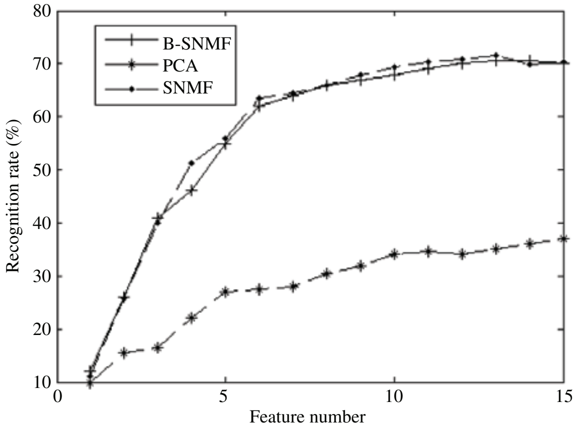


Fig. 5. Recognition rates for B-SNMF, SNMF, and PCA when random occlusions occur in testing images.

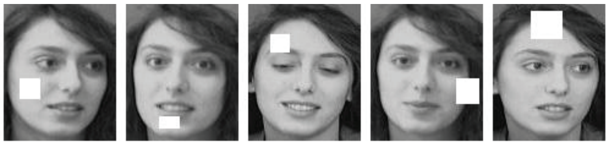


Fig. 6. Examples of occluded face images.

TABLE I

FRACTION OF EASY-EXCLUDED HAAR-LIKE BOX FUNCTIONS OF B-SNMF

Image size	14×12	28×23	44×36	112×92
D	7392	118657	751905	32668188
$\|\Omega\|/D$	61%	47%	37%	25%

TABLE II

NUMBER OF SELECTED HAAR-LIKE BOX FUNCTIONS TO APPROXIMATE 15 BASIS VECTORS

Image Size	14×12	28×23	44×36	112×92
B-PCA	269	424	N/A	N/A
B-SNMF	226	364	407	511

ratio is not larger than 0.1. We observe that B-SNMF utilizes smaller number K of Haar-like box functions than B-PCA does. The basis vector of B-SNMF has compact support and this local property leads to that fewer Haar-like box functions are enough to approximate the basis vector. From (15), one can infer that: the smaller the K is, the faster the feature extraction process will be.

V. CONCLUSION

We have presented a fast part-based subspace selection algorithm, which is termed the binary sparse nonnegative matrix factorization (B-SNMF). B-SNMF makes full use of the sparseness property of the basis vector to remove easy- excluded Haar-like box functions with little computational cost. Besides, the feature extraction process (i.e., testing process) is much faster than binary principal component analysis (B-PCA) without significantly degrades the recognition accuracy.

REFERENCES

[1] X. Li, S. Lin, S. Yan, and D. Xu, "Discriminant locally linear embedding with high-order tensor data," *IEEE Trans. Syst., Man, Cybern., Part B*, vol. 38, no. 2, pp. 342–352, Apr. 2008.

[2] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis: Theory and applications," *ACM Trans. Know. Data Discovery*, vol. 2, no. 3, pp. 1–37, Oct. 2008.

[3] T. Zhang, D. Tao, X. Li, and J. Yang, "Patch alignment for dimensionality reduction," *IEEE Trans. Know. Data Eng.*, Oct. 10, 2008. IEEE computer Society Digital Library.

[4] D. Cai, X. He, and J. Han, "SRDA: An efficient algorithm for large scale discriminant analysis," *IEEE Trans. Know. Data Eng.*, vol. 20, no. 1, pp. 1–12, Jan. 2008.

[5] X. He, S. Yan, Y. Hu, P. Nyogi, and H.-J. Zhang, "Face recognition using laplacianfaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 3, pp. 328–340, Mar. 2005.

[6] J. Ye, Q. Li, H. Xiong, H. Park, R. Janardan, and V. Kumar, "IDR/QR: An incremental dimension reduction algorithm via QR decomposition," *IEEE Trans. Knowledge and Data Eng.*, vol. 17, no. 9, pp. 1208–1222, Sep. 2005.

[7] J. Ye and T. Xiang, "Computational and theoretical analysis of null space and orthogonal linear discriminant analysis," *J. Mach. Learn. Res.*, vol. 7, pp. 1183–1204, 2006.

[8] Y. Fu, S. Yan, and T. S. Huang, "Classification and feature extraction by simplexization," *IEEE Trans. Inf. Forensics Secur.*, vol. 3, no. 1, pp. 91–100, Mar. 2008.

[9] T. Zhang, D. Tao, and J. Yang, "Discriminative locality Alignment," in *Proc. 10th Eur. Conf. Comput. Vision*, Marseille, France, 2008, pp. 725–738.

[10] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric mean for subspace selection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 260–274, Feb. 2009.

[11] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General tensor discriminant analysis and gabor features for gait recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 10, pp. 1700–1715, Oct. 2007.

[12] P. Belhumeur, J. Hespanha, and D. Kriengman, "Eigenfaces vs. fisherfaces: Recognition using class specific linear projections," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 7, pp. 771–720, Jul. 1997.

[13] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Trans. Signal Process.*, vol. 41, no. 12, pp. 3397–3415, Dec. 1993.

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

[15] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, no. 6755, pp. 788–792, Oct. 1999.

[16] M. Turk and A. Pentalnd, "Eigenfaces for recognition," *J. Cognitive Neurosci.*, vol. 3, no. 1, pp. 71–86, Mar. 1991.

[17] F. Tang *et al.*, "Representing images using non-orthogonal haar-like bases," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 29, no. 12, pp. 2120–2134, Dec. 2007.

[18] Y. Pati, R. Rezaiifar, and P. Krishnaprasad, "Orthogonal matching pursuits: Recursive function approximation with applications to wavelet decomposition," in *Proc. 27th Asilomar Conf. Signals, Syst. Comput.*, Pacific Grove, CA, 1993, pp. 40–44.

[19] F. Bennett, T. Richardson, and A. Harter, "Teleporting–making applications mobile," in *Proc. 1994 Workshop Mobile Computing Syst. Applicat.*, Santa Cruz, CA, Dec. 1994, pp. 82–84.

[20] P. Viola and M. Jones, "Rapid object detection using a boosted cascade of simple features," in *Proc. IEEE Int. Conf. Comput. Vision and Pattern Recognition*, Hawaii, vol. 1, 2001, pp. 511–518.

[21] E. Oja, *Subspace Methods of Pattern Recognition.* New York: Wiley, 1983.

[22] D. Tao, X. Li, X. Wu, and S. J. Maybank, "General averaged divergence analysis," in *Proc. IEEE Int. Conf. Data Mining*, Omaha, NE, 2007, pp. 302–311.