Zhijian Yuan and Erkki Oja. A FastICA Algorithm for Non-negative Independent Component Analysis. *In Puntonet, Carlos G.; Prieto, Alberto (Eds.), Proceedings of the Fifth International Symposium on Independent Component Analysis and Blind Signal Separation (ICA 2004)*, Springer Lecture Notes in Computer Science 3195, pp. 1-8, Granada, Spain, 2004.

Projective Nonnegative Matrix Factorization for Image Compression and Feature Extraction

Zhijian Yuan and Erkki Oja

Neural Networks Research Centre,
Helsinki University of Technology,
P.O.Box 5400, 02015 HUT, Finland
{zhijian.yuan, erkki.oja}@hut.fi

Abstract. In image compression and feature extraction, linear expansions are standardly used. It was recently pointed out by Lee and Seung that the positivity or non-negativity of a linear expansion is a very powerful constraint, that seems to lead to sparse representations for the images. Their technique, called Non-negative Matrix Factorization (NMF), was shown to be a useful technique in approximating high dimensional data where the data are comprised of non-negative components. We propose here a new variant of the NMF method for learning spatially localized, sparse, part-based subspace representations of visual patterns. The algorithm is based on positively constrained projections and is related both to NMF and to the conventional SVD or PCA decomposition. Two iterative positive projection algorithms are suggested, one based on minimizing Euclidean distance and the other on minimizing the divergence of the original data matrix and its non-negative approximation. Experimental results show that P-NMF derives bases which are somewhat better suitable for a localized representation than NMF.

1 Introduction

For compressing, denoising and feature extraction of digital image windows, one of the classical approaches is Principal Component Analysis (PCA) and its extensions and approximations such as the Discrete Cosine Transform. In PCA or the related Singular Value Decomposition (SVD), the image is projected on the eigenvectors of the image covariance matrix, each of which provides one linear feature. The representation of an image in this basis is *distributed* in the sense that typically all the features are used at least to some extent in the reconstruction.

Another possibility is a *sparse* representation, in which any given image window is spanned by just a small subset of the available features [1, 2, 6, 10]. This kind of representations have some biological significance, as the sparse features seem to correspond to the receptive fields of simple cells in the area V1 of the mammalian visual cortex. This approach is related to the technique of Independent Component Analysis [3] which can be seen as a nongaussian extension of PCA and Factor Analysis.

H. Kalviainen et al. (Eds.): SCIA 2005, LNCS 3540, pp. 333–342, 2005.

Recently, it was shown by Lee and Seung [4] that *positivity or non-negativity* of a linear expansion is a very powerful constraint that also seems to yield sparse representations. Their technique, called Non-negative Matrix Factorization (NMF), was shown to be a useful technique in approximating high dimensional data where the data are comprised of non-negative components. The authors proposed the idea of using NMF techniques to find a set of basis functions to represent image data where the basis functions enable the identification and classification of intrinsic "parts" that make up the object being imaged by multiple observations. NMF has been typically applied to image and text data [4, 9], but has also been used to deconstruct music tones [8].

NMF imposes the non-negativity constraints in learning the basis images. Both the values of the basis images and the coefficients for reconstruction are all non-negative. The additive property ensures that the components are combined to form a whole in the non-negative way, which has been shown to be the part-based representation of the original data. However, the additive parts learned by NMF are not necessarily localized.

In this paper, we start from the ideas of SVD and NMF and propose a novel method which we call Projective Non-negative Matrix Factorization (P-NMF), for learning spatially localized, parts-based representations of visual patterns. First, in Section 2, we take a look at a simple way to produce a positive SVD by truncating away negative parts. Section 3 briefly reviews Lee's and Seung's NMF. Using this as a baseline, we present our P-NMF method in Section 4. Section 5 gives some experiments and comparisons, and Section 6 concludes the paper.

2 Truncated Singular Value Decomposition

Suppose that our data[1] is given in the form of an $m \times n$ matrix $\mathbf{V}$. Its n columns are the data items, for example, a set of images that have been vectorized by row-by-row scanning. Then m is the number of pixels in any given image. Typically, $n > m$. The Singular Value Decomposition (SVD) for matrix $\mathbf{V}$ is

$$\mathbf{V} = \mathbf{U}\mathbf{D}\hat{\mathbf{U}}^T, \tag{1}$$

where $\mathbf{U}$ $(m \times m)$ and $\hat{\mathbf{U}}$ $(n \times m)$ are orthogonal matrices consisting of the eigenvectors of $\mathbf{V}\mathbf{V}^T$ and $\mathbf{V}^T\mathbf{V}$, respectively, and $\mathbf{D}$ is a diagonal $m \times m$ matrix where the diagonal elements are the ordered singular values of $\mathbf{V}$.

Choosing the r largest singular values of matrix $\mathbf{V}$ to form a new diagonal $r \times r$ matrix $\hat{\mathbf{D}}$, with $r < m$, we get the compressive SVD matrix $\mathbf{X}$ with given rank r,

$$\mathbf{X} = \mathbf{U}\hat{\mathbf{D}}\hat{\mathbf{U}}^T. \tag{2}$$

[1] For clarity, we use here the same notation as in the original NMF theory by Lee and Seung

Now both matrices $\mathbf{U}$ and $\hat{\mathbf{U}}$ have only r columns corresponding to the r largest eigenvalues. The compressive SVD gives the best approximation $\mathbf{X}$ of the matrix $\mathbf{V}$ with the given compressive rank r.

In many real-world cases, for example, for images, spectra etc., the original data matrix $\mathbf{V}$ is *non-negative.* Then the above compressive SVD matrix $\mathbf{X}$ fails to keep the nonnegative property. In order to further approximate it by a non-negative matrix, the following truncated SVD (tSVD) is suggested. We simply truncate away the negative elements by

$$\hat{\mathbf{X}} = \frac{1}{2}(\mathbf{X} + abs(\mathbf{X})). \tag{3}$$

However, it turns out that typically the matrix $\hat{\mathbf{X}}$ in (3) has higher rank than $\mathbf{X}$. Truncation destroys the linear dependences that are the reason for the low rank. In order to get an equal rank, we have to start from a compressive SVD matrix $\mathbf{X}$ with lower rank than the given r. Therefore, to find the truncated matrix $\hat{\mathbf{X}}$ with the compressive rank r, we search all the compressive SVD matrices $\mathbf{X}$ with the rank from 1 to r and form the corresponding truncated matrices. The one with the largest rank that is less than or equal to the given rank r is the truncated matrix $\hat{\mathbf{X}}$ what we choose as the final non-negative approximation. This matrix can be used as a baseline in comparisons, and also as a starting point in iterative improvements. We call this method truncated SVD (t-SVD).

Note that the tSVD only produces the non-negative low-rank approximation $\hat{\mathbf{X}}$ to the data matrix $\mathbf{V}$, but does not give a separable expansion for basis vectors and weights as the usual SVD expansion.

3 Non-negative Matrix Factorization

Given the nonnegative $m \times n$ matrix $\mathbf{V}$ and the constant r, the Nonnegative Matrix Factorization algorithm (NMF) [4] finds a nonnegative $m \times r$ matrix $\mathbf{W}$ and another nonnegative $r \times n$ matrix $\mathbf{H}$ such that they minimize the following optimality problem:

$$\min_{\mathbf{W},\mathbf{H} \geq 0} ||\mathbf{V} - \mathbf{WH}||. \tag{4}$$

This can be interpreted as follows: each column of matrix $\mathbf{W}$ contains a basis vector while each column of H contains the weights needed to approximate the corresponding column in $\mathbf{V}$ using the basis from $\mathbf{W}$. So the product $\mathbf{WH}$ can be regarded as a compressed form of the data in $\mathbf{V}$. The rank r is usually chosen so that $(n+m)r < nm$.

In order to estimate the factorization matrices, an objective function defined by the authors as Kullback-Leibler divergence is

$$\mathbf{F} = \sum_{i=1}^{m} \sum_{\mu=1}^{n} [\mathbf{V}_{i\mu} \log(\mathbf{WH})_{i\mu} - (\mathbf{WH})_{i\mu}]. \tag{5}$$

This objective function can be related to the likelihood of generating the images in $\mathbf{V}$ from the basis $\mathbf{W}$ and encodings $\mathbf{H}$. An iterative approach to reach a local maximum of this objective function is given by the following rules [4, 5]:

$$\mathbf{W}_{ia} \leftarrow \mathbf{W}_{ia} \sum_{\mu} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}} \mathbf{H}_{a\mu}, \mathbf{W}_{ia} \leftarrow \frac{\mathbf{W}_{ia}}{\sum_j \mathbf{W}_{ja}} \tag{6}$$

$$\mathbf{H}_{a\mu} \leftarrow \mathbf{H}_{a\mu} \sum_{i} \mathbf{W}_{ia} \frac{\mathbf{V}_{i\mu}}{(\mathbf{WH})_{i\mu}}. \tag{7}$$

The convergence of the process is ensured[2]. The initialization is performed using positive random initial conditions for matrices $\mathbf{W}$ and $\mathbf{H}$.

4 The Projective NMF Method

4.1 Definition of the Problem

The compressive SVD is a projection method. It projects the data matrix $\mathbf{V}$ onto the subspace of the eigenvectors of the data covariance matrix. Although the truncated method t-SVD outlined above works and keeps nonnegativity, it is not accurate enough for most cases. To improve it, for the given $m \times n$ nonnegative matrix $\mathbf{V}$, $m < n$, let us try to find a subspace $\mathcal{B}$ of R^m, and an $m \times m$ projection matrix $\mathbf{P}$ with given rank r such that $\mathbf{P}$ projects the nonnegative matrix $\mathbf{V}$ onto the subspace $\mathcal{B}$ and keeps the nonnegative property, that is, $\mathbf{PV}$ is a nonnegative matrix. Finally, it should minimize the difference $||\mathbf{V} - \mathbf{PV}||$. This is the basic idea of the Projective NMF method.

We can write any symmetrical projection matrix of rank r in the form

$$\mathbf{P} = \mathbf{W}\mathbf{W}^T \tag{8}$$

with $\mathbf{W}$ an orthogonal $(m \times r)$ matrix[3]. Thus, we can solve the problem by searching for a nonnegative $(m \times r)$ matrix $\mathbf{W}$. Based on this, we now introduce a novel method which we call *Projective Non-negative Matrix Factorization* (P-NMF) as the solution to the following optimality problem

$$\min_{\mathbf{W} \geq 0} ||\mathbf{V} - \mathbf{W}\mathbf{W}^T\mathbf{V}||, \tag{9}$$

where $|| \cdot ||$ is a matrix norm. The most useful norms are the Euclidean distance and the divergence of matrix $\mathbf{A}$ from $\mathbf{B}$, defined as follows: The Euclidean distance between two matrices $\mathbf{A}$ and $\mathbf{B}$ is

[2] The matlab program for the above update rules is available at `http://journalclub.mit.edu` under the "Computational Neuroscience" discussion category.

[3] This is just notation for a generic basis matrix; the solution will not be the same as the $\mathbf{W}$ matrix in NMF.

$$||\mathbf{A}-\mathbf{B}||^2=\sum_{i,j}(\mathbf{A}_{ij}-\mathbf{B}_{ij})^2, \tag{10}$$

and the divergence of **A** from **B**

$$D(\mathbf{A}||\mathbf{B})=\sum_{i,j}(\mathbf{A}_{ij}\log\frac{\mathbf{A}_{ij}}{\mathbf{B}_{ij}}-\mathbf{A}_{ij}+\mathbf{B}_{ij}). \tag{11}$$

Both are lower bounded by zero, and vanish if and only if $\mathbf{A}=\mathbf{B}$.

4.2 Algorithms

We first consider the Euclidean distance (10). Define the function

$$\mathbf{F}=\frac{1}{2}||\mathbf{V}-\mathbf{W}\mathbf{W}^T\mathbf{V}||^2. \tag{12}$$

Then the unconstrained gradient of **F** for **W**, $\frac{\partial\mathbf{F}}{\partial\mathbf{w}_{ij}}$, is given by

$$\frac{\partial\mathbf{F}}{\partial\mathbf{w}_{ij}}=-2(\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}+(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}+(\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}. \tag{13}$$

Using the gradient we can construct the additive update rule for minimization,

$$\mathbf{W}_{ij}\leftarrow\mathbf{W}_{ij}-\eta_{ij}\frac{\partial\mathbf{F}}{\partial\mathbf{w}_{ij}} \tag{14}$$

where η_{ij} is the positive step size.

However, there is nothing to guarantee that the elements $\mathbf{W}_{ij}$ would stay non-negative. In order to ensure this, we choose the step size as follows,

$$\eta_{ij}=\frac{\mathbf{W}_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}+(\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}}. \tag{15}$$

Then the additive update rule (14) can be formulated as a multiplicative update rule,

$$\mathbf{W}_{ij}\leftarrow\mathbf{W}_{ij}\frac{(\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}+(\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}}. \tag{16}$$

Now it is guaranteed that the $\mathbf{W}_{ij}$ will stay nonnegative, as everything on the right-hand side is nonnegative.

For the divergence measure (11), we follow the same process. First we calculate the gradient

$$\frac{\partial D(\mathbf{V}||\mathbf{W}\mathbf{W}^T\mathbf{V})}{\partial\mathbf{w}_{ij}}=\sum_k\left((\mathbf{W}^T\mathbf{V})_{jk}+\sum_l\mathbf{W}_{lj}\mathbf{V}_{ik}\right) \tag{17}$$

$$-\sum_k\mathbf{V}_{ik}(\mathbf{W}^T\mathbf{V})_{jk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} \tag{18}$$

$$-\sum_k\mathbf{V}_{ik}\sum_l\mathbf{W}_{lj}\mathbf{V}_{lk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{lk}. \tag{19}$$

Using the gradient, the additive update rule becomes

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} + \zeta_{ij} \frac{\partial D(\mathbf{V}||\mathbf{W}\mathbf{W}^T\mathbf{V})}{\partial \mathbf{w}_{ij}} \tag{20}$$

where ζ_{ij} is the step size. Choosing this step size as following,

$$\zeta_{ij} = \frac{\mathbf{W}_{ij}}{\sum_k \mathbf{V}_{ik}\left[(\mathbf{W}^T\mathbf{V})_{jk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{lk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{lk}\right]}. \tag{21}$$

we obtain the multiplicative update rule

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{\sum_k \left((\mathbf{W}^T\mathbf{V})_{jk} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{ik}\right)}{\sum_k \mathbf{V}_{ik}((\mathbf{W}^T\mathbf{V})_{jk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{lk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{lk})}. \tag{22}$$

It is easy to see that both multiplicative update rules (16) and (22) can ensure that the matrix $\mathbf{W}$ is non-negative.

4.3 The Relationship Between NMF and P-NMF

There is a very obvious relationship between our P-NMF algorithms and the original NMF. Comparing the two optimality problems, P-NMF (9) and the original NMF (4), we see that the weight matrix $\mathbf{H}$ in NMF is simply replaced by $\mathbf{W}^T\mathbf{V}$ in our algorithms. Both multiplicative update rules (16) and (22) are obtained similar to Lee and Seung's algorithms [5]. Therefore, the convergence of these two algorithms can also be proved following Lee and Seung [5] by noticing that the coefficient matrix $\mathbf{H}$ is replaced by $\mathbf{W}\mathbf{V}$.

4.4 The Relationship Between SVD and P-NMF

There is also a relationship between the P-NMF algorithm and the SVD. For the Euclidean norm, note the similarity of the problem (9) with the conventional PCA for the columns of $\mathbf{V}$. Removing the positivity constraint, this would become the usual finite-sample PCA problem, whose solution is known to be an orthogonal matrix consisting of the eigenvectors of $\mathbf{V}\mathbf{V}^T$. But this is the matrix $\mathbf{U}$ in the SVD of eq. (1). However, now with the positivity constraint in place, the solution will be something quite different.

5 Simulations

5.1 Data Preparation

As experimental data, we used face images from the MIT-CBCL database and derived the NMF and P-NMF expansions for them. The training data set contains 2429 faces. Each face has $19 \times 19 = 361$ pixels and has been histogram-equalized and normalized so that all pixel values are between 0 and 1. Thus the data matrix $\mathbf{V}$ which now has the faces as columns is 361 × 2429. This matrix was compressed to rank $r = 49$ using either t-SVD, NMF, or P-NMF expansions.

5.2 Learning Basis Components

The basis images of tSVD, NMF, and P-NMF with dimension 49 are shown in Figure 1. For NMF and P-NMF, these are the 49 columns of the corresponding matrices $\mathbf{W}$. For t-SVD, we show the 49 basis vectors of the range space of the rank-49 nonnegative matrix $\hat{\mathbf{X}}$, obtained by ordinary SVD of this matrix. Thus the basis images for NMF and P-NMF are truly non-negative, while the t-SVD only produces a non-negative overall approximation to the data but does not give a separable expansion for basis vectors and weights.

All the images are displayed with the matlab command "imagesc" without any extra scale. Both NMF and P-NMF bases are holistic for the training set. For this problem, the P-NMF algorithm converges about 5 times faster than NMF.

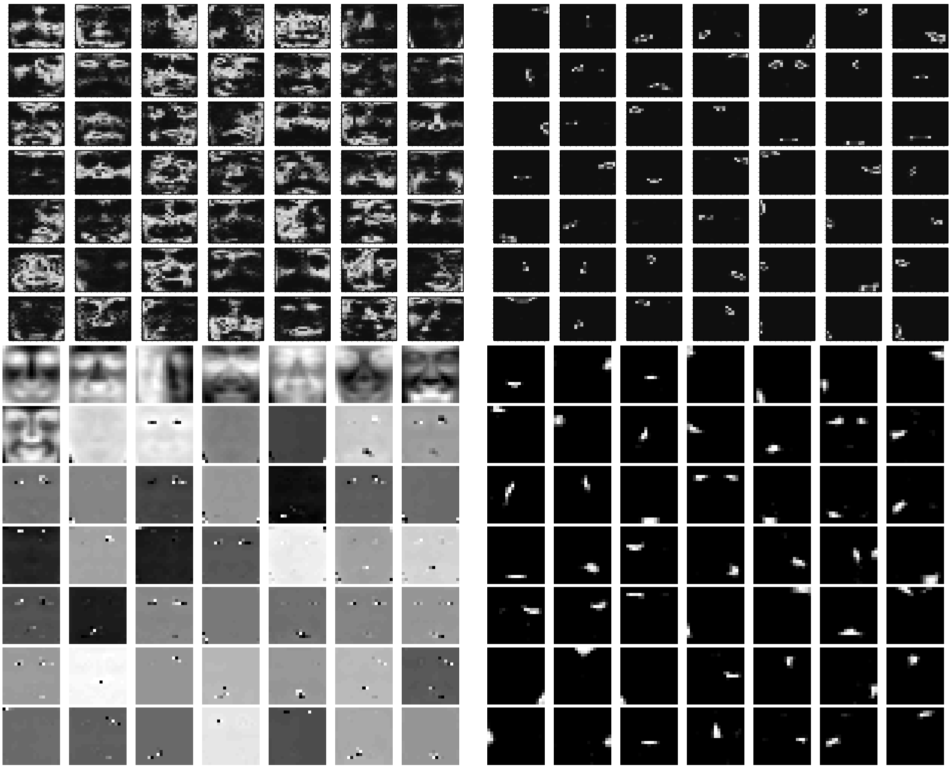

Fig. 1. NMF (top, left), t-SVD (bottom, left) and the two versions of the new P-NMF method (right) bases of dimension 49. Each basis component consists of 19 × 19 pixels

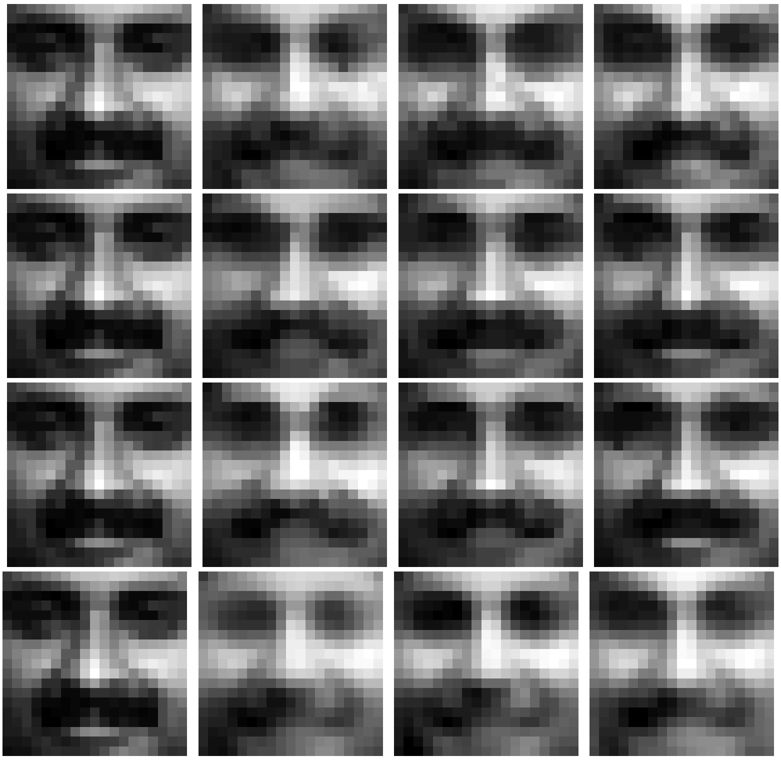

Fig. 2. The original face image (left) and its reconstructions by NMF (top row), the two versions of the new P-NMF method under 100 iterative steps (second and third rows), and t-SVD (bottom row). The dimensions in columns 2, 3, and 4 are 25, 49 and 81, respectively

5.3 Reconstruction Accuracy

We repeated the above computations for ranks $r = 25, 49$ and 81. Figure 2 shows the reconstructions for one of the face images in the t-SVD, NMF, and P-NMF subspaces of corresponding dimensions. For comparison, also the original face image is shown. As the dimension increases, more details are recovered. Visually, the P-NMF method is comparable to NMF.

The recognition accuracy, defined as the Euclidean distance between the original data matrix and the recognition matrix, can be used to measure the performance quantitatively. Figure 3 shows the recognition accuracy curves of P-NMF and NMF under different iterative steps. NMF converges faster, but when the number of steps increases, P-NMF works very similarly to NMF. One thing to be noticed is that the accuracy of P-NMF depends on the initial values. Although the number of iteration steps is larger in P-NMF for comparable error with NMF, this is compensated by the fact that the computational complexity for one iteration step is considerably lower for P-NMF, as only one matrix has to be updated instead of two.

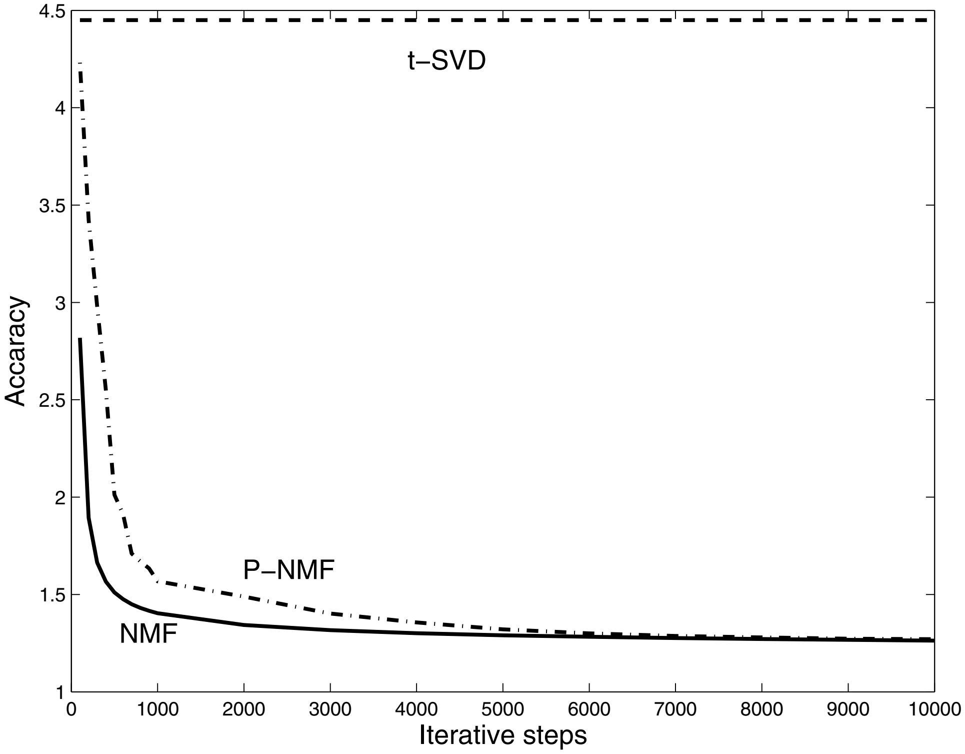

Fig. 3. Recognition accuracies (unit: 10^8) versus iterative steps using t-SVD, NMF and P-NMF with compressive dimension 49

6 Conclusion

We proposed a new variant of the well-known Non-negative Matrix Factorization (NMF) method for learning spatially localized, sparse, part-based subspace representations of visual patterns. The algorithm is based on positively constrained projections and is related both to NMF and to the conventional SVD decomposition. Two iterative positive projection algorithms were suggested, one based on minimizing Euclidean distance and the other on minimizing the divergence of the original data matrix and its approximation. Compared to the NMF method, the iterations are somewhat simpler as only one matrix is updated instead of two as in NMF. The tradeoff is that the convergence, counted in iteration steps, is slower than in NMF.

One purpose of these approaches is to learn localized features which would be suitable not only for image compression, but also for object recognition. Experimental results show that P-NMF derives bases which are better suitable for a localized representation than NMF. It remains to be seen whether they would be better in pattern recognition, too.

References

1. A. Bell and T. Sejnowski. The "independent components" of images are edge filters. *Vision Research*, 37: 3327–3338, 1997.
2. A. Hyvärinen and P. Hoyer. Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 13: 1527–1558, 2001.
3. A. Hyvärinen, J. Karhunen, and E. Oja. *Independent Component Analysis*. Wiley, New York, 2001.
4. D. D. Lee and H. S. Seung. Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
5. D. D. Lee and H. S. Seung. Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562, 2000.
6. B. A. Olshausen and D. J. Field. Natural image statistics and efficient coding. *Network*, 7: 333–339, 1996.
7. P. Paatero and U. Tapper. Positive Matrix Factorization: A non-negative factor model with optimal utilization of error estimations of data values. *Environmetrics*, 5, 111-126, 1997.
8. T. Kawamoto, K. Hotta, T. Mishima, J. Fujiki, M. Tanaka and T. Kurita. Estimation of single tones from chord sounds using non-negative matrix factorization. *Neural Network World*, 3, 429-436, July 2000.
9. L.K. Saul and D.D. Lee. Multiplicative updates for classification by mixture modela. In *Advances in Neural Information Processing Systems* 14, 2002.
10. J. H. van Hateren and A. van der Schaaf. Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. Royal Soc. London B*, 265: 2315–2320, 1998.