Noname manuscript No.
(will be inserted by the editor)

Projective Nonnegative Matrix Factorization: Sparseness, Orthogonality, and Clustering

Zhijian Yuan · Zhirong Yang · Erkki Oja

Received: date / Accepted: date

Abstract

In image compression and feature extraction, linear expansions are standardly used. It was 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 useful in approximating high dimensional data where the data are comprised of non-negative components. We have earlier proposed a new variant of the NMF method, called Projective Nonnegative Matrix Factorization, 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. In this paper we show that PNMF is intimately related to "soft" k-means clustering and is able to outperform NMF in document classification tasks. The reason is that PNMF derives bases which are somewhat better for a localized representation than NMF, more orthogonal, and produce considerably more sparse representations.

Keywords Projective Nonnegative Matrix Factorization · Sparseness · Orthogonality · Clustering

1 Introduction

For compressing, denoising and feature extraction of data sets such as digital image collections, term-document matrixes for text, spectra, etc., one of the classical approaches is Principal Component Analysis (PCA). In PCA or the related Singular Value Decomposition (SVD) [3], each data vector is projected on the eigenvectors of the covariance matrix, each of which provides one linear feature. The representation of data in this basis is *distributed* in the sense that typically all the features are used at least to some extent in the reconstruction.

Adaptive Informatics Research Centre
Helsinki University of Technology
P.O.Box 5400, 02015 HUT, Finland
E-mail: {zhijian.yuan, zhirong.yang, erkki.oja}@hut.fi

Another possibility is a *sparse* representation, in which any given data item is spanned by just a small subset of the available features [1,9,11,15,20]. It was shown by Lee and Seung [12] that *positivity or non-negativity* of a linear expansion is a very powerful constraint that 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 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 and separately learned in an iterative process. The additive property ensures that the components are combined to form a whole in the non-negative way, which has been shown to lead to the part-based representation of the original data. However, the additive parts learned by NMF are not necessarily localized.

NMF has been typically applied to image and text data [7,16], but has also been used to deconstruct music tones [18,19] and for spectral data analysis [17]. The close relation of NMF to clustering has been shown by Ding *et al* [4]. For recent developments in NMF, see [2] and references therein.

In [23], starting from the ideas of SVD and NMF, we proposed a novel method which we call Projective Non-negative Matrix Factorization (PNMF), for learning spatially localized, parts-based representations of visual patterns. PNMF uses only one non-negative matrix in the expansion, instead of the two matrices used in NMF, and thus has much less free parameters to be learned. Learning rules were given and it was shown that for face images, PNMF produces spatially more localized and non-overlapping basic components than NMF. One reason for this turned out to be that the basic vectors of PNMF are clearly more orthogonal than those of NMF [21]. More thorough analysis of the learning rules was also given in [21,22].

The present paper gives several extensions to the basic ideas. First, in Section 2, we take a look at a very simple way to produce a positive SVD by truncating away negative parts, which may serve as the initial point for the new PNMF learning algorithm. Section 3 briefly reviews Lee's and Seung's NMF. Using this as a baseline, we briefly review our PNMF method in Section 4. Section 5 shows that PNMF is even more closely related to clustering than NMF: one of the cost functions of PNMF is in fact exactly equal to the usual cost function of k-means clustering, except for the constraints that are used in both. Results on document data clustering are given, showing that in these experiments PNMF indeed gives somewhat better accuracy and entropy than NMF and the classical k-means clustering. Section 6 gives experimental results on the sparseness and orthogonality of PNMF basis functions, and Section 7 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

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

Singular Value Decomposition (SVD) for matrix $\mathbf{V}$ is

$$\mathbf{V} = \mathbf{QDR}^T, \tag{1}$$

where $\mathbf{Q}$ $(m \times m)$ and $\mathbf{R}$ $(n \times m)$ are orthogonal matrices[2] consisting of the eigenvectors of $\mathbf{VV}^T$ 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 $\mathbf{U}$ with given rank r,

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

Now both eigenvector matrices $\hat{\mathbf{Q}}$ and $\hat{\mathbf{R}}$ have only r columns, corresponding to the r largest eigenvalues. The compressive SVD gives the best approximation (in Frobenius matrix norm) of the matrix $\mathbf{V}$ with the given compressive rank r [6].

In the case that we consider here, all the elements of the data matrix $\mathbf{V}$ are *non-negative*. Then the above compressive SVD matrix $\mathbf{U}$ 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

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

where the absolute value is taken element by element.

However, it turns out that typically the matrix $\hat{\mathbf{U}}$ in (3) has higher rank than $\mathbf{U}$. 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 $\mathbf{U}$ with lower rank than the given r. To find the truncated matrix $\hat{\mathbf{U}}$ with the compressive rank r, we search all the compressive SVD matrices $\mathbf{U}$ 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 $\hat{\mathbf{U}}$ 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 (tSVD).

Note that the tSVD only produces the non-negative low-rank approximation $\hat{\mathbf{U}}$ to the data matrix $\mathbf{V}$, but does not give a separable expansion for basis vectors and weights, like 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) [12] 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 $\mathbf{H}$ contains the weights needed to approximate the corresponding column in $\mathbf{V}$ using those basis vectors.

[2] In the usual definition of SVD, $\mathbf{R}$ is a full $n \times n$ matrix, but the result is the same as there are only m nonzero eigenvectors

In order to estimate the factorization matrices, an objective function defined in [12] is the Kullback-Leibler divergence

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

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 [12,13]:

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

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

The convergence of the process is ensured. 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

In NMF, the two matrices $\mathbf{W}$ and $\mathbf{H}$ contain a total of $r\times(m+n)$ free parameters. This gives a certain ambiguity to the problem. For example, consider the simplest possible case in which $\mathbf{V}$ and $\mathbf{W}$ are just column vectors and $\mathbf{H} = H$ is a scalar: obviously there are an infinite number of solutions $\mathbf{W} = \frac{1}{H}\mathbf{V}$ with H arbitrary.

In [23] we presented a modification of NMF that contains only $r\times m$ free parameters. Thus, the number of parameters is always less than or equal to the number of elements in the data matrix $\mathbf{V}$. The modification is based on an approximative projection. As the starting point, consider the compressive SVD which is a projection method. It projects the nonnegative $m \times n$ data matrix $\mathbf{V}$ onto the subspace of the first r eigenvectors of the data covariance matrix – formally, eqs. (1) and (2) give

$$\mathbf{U} = \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{V}.$$

Matrix $\hat{\mathbf{Q}}\hat{\mathbf{Q}}^T$ is the projection matrix on the eigenvector subspace. This is the unconstrained optimal approximation to $\mathbf{V}$ in the space of rank r matrices: :

$$||\mathbf{V} - \hat{\mathbf{Q}}\hat{\mathbf{Q}}^T\mathbf{V}|| = minimum. \tag{8}$$

Generally, matrix $\hat{\mathbf{Q}}$ is not nonnegative.

To improve on this, let us try to find a *nonnegative* $m\times m$ approximative projection matrix $\mathbf{P}$ with given rank r, which minimizes the difference $||\mathbf{V} - \mathbf{PV}||$. We can write any symmetrical projection matrix of rank r in the form

$$\mathbf{P} = \mathbf{WW}^T \tag{9}$$

with $\mathbf{W}$ an orthogonal $(m \times r)$ matrix[3].

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

Based on this, we introduced [23] a novel method which we call *Projective Nonnegative Matrix Factorization* (PNMF) as the solution to the following optimality problem

$$\min_{\mathbf{W}\geq 0} ||\mathbf{V} - \mathbf{W}\mathbf{W}^T\mathbf{V}||, \tag{10}$$

where $||\cdot||$ is a matrix norm.

The most useful norm is the Euclidean distance between two matrices $\mathbf{A}$ and $\mathbf{B}$, or the Frobenius matrix norm of their difference:

$$||\mathbf{A} - \mathbf{B}||^2 = \sum_{i,j}(\mathbf{A}_{ij} - \mathbf{B}_{ij})^2. \tag{11}$$

Another possibility that will be considered, in analogy with NMF, is the divergence[4] of matrix $\mathbf{A}$ from $\mathbf{B}$, defined as

$$D(\mathbf{A}||\mathbf{B}) = \sum_{i,j}(\mathbf{A}_{ij}\log\frac{\mathbf{A}_{ij}}{\mathbf{B}_{ij}} - \mathbf{A}_{ij} + \mathbf{B}_{ij}). \tag{12}$$

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

The PNMF method seems to offer some advantages as compared to NMF. The first one is increased orthogonality of the basis vectors. This is due to the similarity of the criterion (10) to SVD. Removing the positivity constraint but keeping the rank constraint, an orthogonal eigenvector basis is the solution. For positive bases, orthogonality is intimately connected to sparseness.

Second, consider the case in which the $\mathbf{V}$ matrix is just a training set and the goal is to find the representation not only for the columns of $\mathbf{V}$ but for new vectors, too. For PNMF, the representation for any column of $\mathbf{V}$, say $\mathbf{v}$, is simply $\mathbf{W}\mathbf{W}^T\mathbf{v}$ and that can be easily computed for a new vector, too. In NMF, there is no such natural representation because both $\mathbf{W}$ and $\mathbf{H}$ are needed, and matrix $\mathbf{H}$ has only n columns. The extra column in $\mathbf{H}$ would have to be recomputed from the criterion.

Third, as pointed out by [4,14], NMF has a close relation to clustering. The relation of PNMF to clustering is even closer, as shown below in Section 5. It turns out that PNMF provides a novel way to perform "soft" k-means clustering on a dataset.

4.2 Learning algorithms

We first consider the Euclidean distance (11). Define the function

$$\mathbf{F}(\mathbf{W}) = \frac{1}{2}||\mathbf{V} - \mathbf{W}\mathbf{W}^T\mathbf{V}||^2. \tag{13}$$

Before developing the algorithm, we need the following lemma.

Lemma 1 *For the given matrices* $\mathbf{W}$ *and* $\mathbf{V}$*, the minimization of* $f(\lambda) = ||\mathbf{V} - \lambda\mathbf{W}\mathbf{W}^T\mathbf{V}||^2$ *corresponding to* λ *is reached at*

$$\lambda = \frac{trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T]}{trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T]}. \tag{14}$$

[4] Formally, this is not a norm or metric

Proof. By setting $\frac{\partial f(\lambda)}{\partial \lambda} = 0$:

$$\frac{\partial f(\lambda)}{\partial \lambda} = trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T] - \lambda trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T], \tag{15}$$

we obtain

$$\lambda = \frac{trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T]}{trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T]}. \tag{16}$$

Let us now calculate the unconstrained gradient of $\mathbf{F}$ for $\mathbf{W}$, $\frac{\partial \mathbf{F}}{\partial \mathbf{w}_{ij}}$, which is given by

$$\frac{\partial \mathbf{F}}{\partial \mathbf{w}_{ij}} = -2(\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}. \tag{17}$$

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

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} - \eta_{ij}\frac{\partial \mathbf{F}}{\partial \mathbf{w}_{ij}} \tag{18}$$

where η_{ij} is the positive step size.

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

$$\eta_{ij} = \frac{\mathbf{W}_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}}. \tag{19}$$

Then the additive update rule (18) can be formulated as a multiplicative update rule,

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij}\frac{2(\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij}}{(\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W})_{ij} + (\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T\mathbf{W})_{ij}}. \tag{20}$$

and followed by normalizing the above updated matrix $\mathbf{W}$ to keep the basis vectors close to the unit sphere.

$$\mathbf{W} \leftarrow \mathbf{W}\sqrt{trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T]/trace[\mathbf{W}\mathbf{W}^T\mathbf{V}\mathbf{V}^T\mathbf{W}\mathbf{W}^T]}. \tag{21}$$

Now it is guaranteed that the $\mathbf{W}_{ij}$ will stay nonnegative, as everything on the right-hand side is nonnegative. It is worth to notice that when the matrix $\mathbf{W}$ is a vector, then the equation (21) is the general normanization.

For the divergence measure (12), we follow the same process.

Lemma 2 *For the given matrices* $\mathbf{W}$ *and* $\mathbf{V}$*, the minimization of* $f(\lambda) = \mathbf{D}(\mathbf{V}||\lambda\mathbf{W}\mathbf{W}^T\mathbf{V})$ *corresponding to* λ *is reached at*

$$\lambda = \frac{\sum_{ij}\mathbf{V}_{ij}}{\sum_{ij}(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij}}. \tag{22}$$

Proof. Again, by setting $\frac{\partial f(\lambda)}{\partial \lambda} = 0$:

$$\frac{\partial f(\lambda)}{\partial \lambda} = \sum_{ij} (\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij} - \sum_{ij} \mathbf{V}_{ij}/\lambda = 0, \tag{23}$$

the lemma is proven.

The gradient of $D(\mathbf{V}||\mathbf{W}\mathbf{W}^T\mathbf{V})$ to $\mathbf{W}$ is

$$\frac{\partial D(\mathbf{V}||\mathbf{W}\mathbf{W}^T\mathbf{V})}{\partial \mathbf{w}_{ij}} = \sum_k \left((\mathbf{W}^T\mathbf{V})_{jk} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{ik} \right) \tag{24}$$

$$- \sum_k \mathbf{V}_{ik}(\mathbf{W}^T\mathbf{V})_{jk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} \tag{25}$$

$$- \sum_k \mathbf{V}_{ik} \sum_l \mathbf{W}_{lj}\mathbf{V}_{lk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{lk}. \tag{26}$$

Using the gradient, the additive update rule becomes

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} - \zeta_{ij} \frac{\partial D(\mathbf{V}||\mathbf{W}\mathbf{W}^T\mathbf{V})}{\partial \mathbf{w}_{ij}} \tag{27}$$

where ζ_{ij} is the step size. Choosing this step size as follows:

$$\zeta_{ij} = \frac{\mathbf{W}_{ij}}{\sum_k \left((\mathbf{W}^T\mathbf{V})_{jk} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{ik} \right)}. \tag{28}$$

we obtain the multiplicative update rule

$$\mathbf{W}_{ij} \leftarrow \mathbf{W}_{ij} \frac{\sum_k \mathbf{V}_{ik} \left((\mathbf{W}^T\mathbf{V})_{jk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{ik} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{lk}/(\mathbf{W}\mathbf{W}^T\mathbf{V})_{lk} \right)}{\sum_k \left((\mathbf{W}^T\mathbf{V})_{jk} + \sum_l \mathbf{W}_{lj}\mathbf{V}_{ik} \right)}, \tag{29}$$

followed by "normalizing" the above updated matrix $\mathbf{W}$

$$\mathbf{W} \leftarrow \mathbf{W} \sqrt{\frac{\sum_{ij} \mathbf{V}_{ij}}{\sum_{ij} (\mathbf{W}\mathbf{W}^T\mathbf{V})_{ij}}}. \tag{30}$$

It is easy to see that both multiplicative update rules (20) and (29) will ensure that the matrix $\mathbf{W}$ is non-negative. The convergence of the two algorithms is complicated, and the full proof has not been done, yet.

4.3 The relationship between NMF and PNMF

There is a very obvious relationship between our PNMF algorithms and the original NMF. Comparing the two optimality problems, PNMF (10) 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 (20) and (29) are similar to Lee and Seung's algorithms [13]. The number of free parameters is much smaller in PNMF.

4.4 The relationship between SVD and PNMF

There is also a relationship between the PNMF algorithm and the SVD. For the Euclidean norm, note the similarity of the problem (10) 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 $\mathbf{Q}$ in the SVD of eq. (1). However, now with the positivity constraint in place, the solution will be something quite different.

5 Relation to *k*-means clustering

It is well-known that k-means clustering is related to nonnegative factorizations [4]. Assume we want to cluster a set of n-dimensional vectors $\mathbf{x}_1, ..., \mathbf{x}_m$ into k clusters $C_1, ..., C_k$. The classical k-means clustering uses k cluster centroids $\mathbf{m}_1, ..., \mathbf{m}_k$ to characterize the clusters. The objective function is

$$J_k = \sum_{j=1}^{k} \sum_{i \in C_j} \|\mathbf{x}_i - \mathbf{m}_j\|^2. \tag{31}$$

As shown by Ding et al [4,14], this can be written as

$$J_k = trace[\mathbf{X}^T\mathbf{X}] - trace[\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H}] \tag{32}$$

with $\mathbf{X} = (\mathbf{x}_1, , \mathbf{x}_m)$ the data matrix, and $\mathbf{H}$ the *indicator matrix* for the clusters: $\mathbf{H}_{ij} = 1$ if vector $\mathbf{x}_i$ belongs to cluster C_j, zero otherwise. Thus $\mathbf{H}$ is a binary $(m \times k)$ matrix, whose columns are orthogonal, because each vector belongs to one and only one cluster. Minimizing J_k under the binary and orthogonality constraints on $\mathbf{H}$ is equivalent to maximizing $trace[\mathbf{H}^T\mathbf{X}^T\mathbf{X}\mathbf{H}]$ under these constraints.

The PNMF has a direct relation to this. Consider the PNMF criterion for the *transposed* data matrix $\mathbf{X}^T$:

$$\|\mathbf{X}^T - \mathbf{W}\mathbf{W}^T\mathbf{X}^T\|^2 = trace[(\mathbf{X}^T - \mathbf{W}\mathbf{W}^T\mathbf{X}^T)(\mathbf{X} - \mathbf{X}\mathbf{W}\mathbf{W}^T)] \tag{33}$$

$$= trace[\mathbf{X}^T\mathbf{X}] - 2trace[\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}] + trace[\mathbf{W}\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}\mathbf{W}^T]. \tag{34}$$

Assuming that the columns of $\mathbf{W}$ were *orthonormal*, i.e., $\mathbf{W}^T\mathbf{W} = \mathbf{I}$, the last term becomes $trace[\mathbf{W}^T\mathbf{X}^T\mathbf{X}\mathbf{W}]$ and the whole PNMF criterion becomes exactly equal to the k-means criterion J_k in eq. (32), except for the binary constraint.

PNMF can thus be used for clustering the columns of a data matrix $\mathbf{X}$ in the following way: apply PNMF for the transposed matrix $\mathbf{X}^T$ (which is now $(m \times n)$ like in the original formulation of PNMF) under the usual non-negativity constraint and obtain the weight matrix $\mathbf{W}$. The rank r of $\mathbf{W}$ should be set equal to k, the desired number of clusters. If necessary, the columns of $\mathbf{W}$ may be normalized to unit length. As the analysis above shows, $\mathbf{W}$ is then an approximation of the cluster indicator matrix, whose m rows correspond to the m data vectors to be clustered, and k columns correspond to the k clusters: the elements $\mathbf{W}_{ij},\ j = 1, ..., k$ along the i-th row show a "soft" clustering of the i-th data vector into the clusters $C_j,\ j = 1, ..., k$. Correspondingly, the same elements along the j-th column show the degrees by which each of the data vectors belongs to the j-th cluster C_j. Because of the constraint, all

these degrees are non-negative. If a unique "hard" clustering is desired, the maximum element on each row can be chosen to indicate the cluster.

Also the matrix $\mathbf{XW}$ whose transpose $\mathbf{W}^T\mathbf{X}^T$ appears in (33) has a very clear interpretation: its columns directly give the "soft" cluster centroids $\mathbf{m}_1, ..., \mathbf{m}_k$. Namely, the j-th column of $\mathbf{XW}$ equals $\sum_{i=1}^{m} \mathbf{W}_{ij}\mathbf{x}_i$, thus giving the weighted average of the data vectors, weighted according to how much they belong to cluster C_j. If the elements were binary as is the case in the hard k-means clustering, then these columns would be exactly the cluster centroids $\mathbf{m}_j$. In the "soft" clustering given by PNMF, they are still the optimal cluster mean vectors.

For this clustering scheme to be valid, an essential question is how good an approximation $\mathbf{W}$ will be to the binary indicator matrix $\mathbf{H}$, given that the constraint used in PNMF is just the non-negativity of $\mathbf{W}$. This is where the *sparsity* and the good *orthogonality* properties of $\mathbf{W}$ may come into play. As shown in Section 6.2, PNMF is able to produce a clearly more orthogonal matrix than NMF (see Figure 3), and thus can be expected to produce a better clustering result than NMF or its variants. Some experimental clustering results are given in the following.

6 Simulations

6.1 Document clustering

We use three datasets: 20 Newsgroups dataset, MEDLINE dataset and Reuters in our experiments. They are frequently used in the information retrieval research.

MEDLINE consists of 1033 abstracts from medical journals. In the MEDLINE dataset, there are 30 natural language queries and relations giving relevance judgements between query and document. We prepared a term-document matrix of size 5735×696 since only 696 documents among the 1033 documents have matched with the 30 queries. The 20 Newsgroups data set is a collection of 20000 messages taken from 20 newsgroups. We use a subset of the Newsgroup data which contains 100 randomly selected messages from each newsgroup. The Reuters-21578 Text Categorization Test Collection contains documents collected from the Reuters newswire in 1987. In our experiments, we use a subset of the data collection which includes the 10 most frequent categories among the 135 topics and has about 2900 documents.

To measure the clustering performance, we use accuracy and entropy as our performance measures as defined in [5]. In these experiments, binary clustering is used, achieved by locating the maximum elements of the basis matrix $\mathbf{W}$. Accuracy discovers the one-to-one relationship between clusters and classes and measures the extent to which each cluster contained data points from the corresponding class. It sums up the whole matching degree between all pairs of classes and clusters. Accuracy can be represented as:

$$Accuracy = \max\left(\sum_{i,j} T(C_i, L_j)\right)/N \tag{35}$$

where C_i is the i-th cluster, and L_j is the j-th class. $T(C_i, L_j)$ is the number of entities which belong to class j and are assigned to cluster i. Accuracy computes the maximum sum of $T(C_i, L_j)$ for all pairs of clusters and classes, and these pairs have no overlaps. Generally, the greater the accuracy value, the better the clustering performance.

Entropy measures how classes are distributed on various clusters. Generally, the smaller the entropy value, the better the clustering quality. Following [5], the entropy of the entire clustering solution is computed as:

$$Entropy = -\frac{1}{n \log_2 m} \sum_{i=1}^{k} \sum_{j=1}^{c} n_i^j \log_2 \frac{n_i^j}{n_i} \tag{36}$$

where c is the number of original categories, k is the number of clusters, n_i is the size of cluster i, and n_i^j gives the number of points in cluster j that belong to the i-th category. Generally, the smaller the entropy value, the better the clustering quality.

For each of the three datasets we run k-means clustering, NMF and PNMF for a comparison. The clustering solutions of NMF and PNMF are compared based on accuracy and entropy as shown in Figs. 1 and 2.

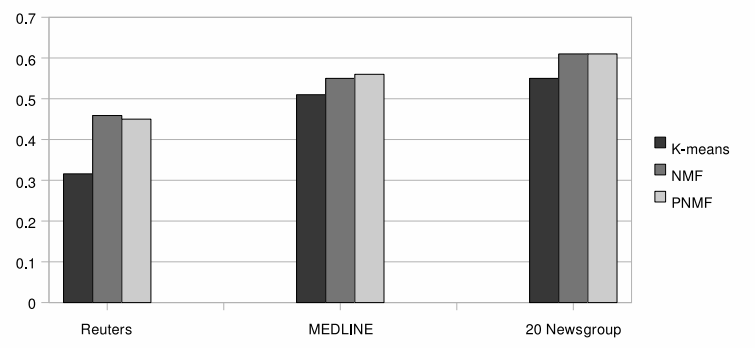

Fig. 1 Accuracies of K-means, NMF and PNMF.

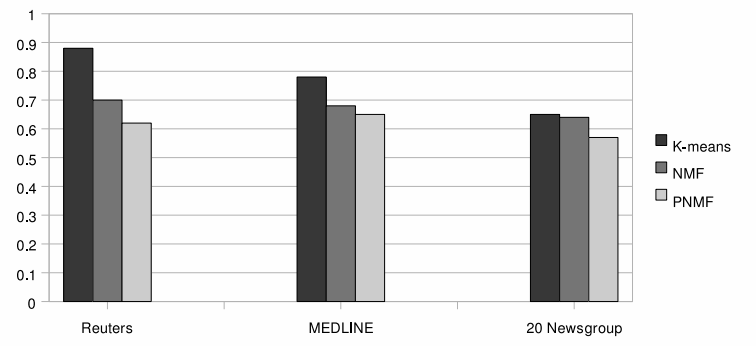

Fig. 2 Entropies of K-means, NMF and PNMF.

Figure 1 shows that NMF and PNMF have similar performance on accuracy, and Figure 2 shows that PNMF has lower entropy value than NMF which means it has better clustering quality.

6.2 Facial image data: orthogonality and sparseness

In this experiment, we employed Lee's NMF algorithm [12], Hoyer's NNSC [8] algorithm and our PNMF methods for image compression, comparing their performance with respect to sparseness and orthogonality. We used face images from the MIT-CBCL database as experimental data, and derived the NMF, NNSC and PNMF 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 tSVD, NMF, NNSC or PNMF expansions. The visual results have been shown in our previous papers [23, 21]. Here, we will give some quantitative analysis on the localization and sparseness. Define entropy for each of the 49 normalized columns of the basis matrix $\mathbf{W}$ (the basis images) as

$$en_j = -\sum_{1}^{361} \mathbf{W}_{ij} \log \mathbf{W}_{ij}, \sum_{1}^{361} \mathbf{W}_{ij} = 1, \tag{37}$$

then calculate the average of entropies over the 49 basis images. Generally, a smaller entropy value shows more localization and sparseness. Computing the average entropies of the basis images derived by NMF, NNSC, tSVD, and PNMF with Euclidean and divergence measures, gives the values 22.329, 22.671, 54.528, 8.5179 and 7.3534, respectively. Thus the two versions of PNMF have clearly the smallest entropy, hence sparseness for the basis images.

Another way to measure the sparseness is the orthogonality of the basis vectors, since two nonnegative vectors are orthogonal if and only if they do not have the same non-zero elements. Therefore the orthogonality between the learned bases reveals the sparsity of the resulting representations, and the amount of localization for facial images. We measure the orthogonality of the learned bases by the following

$$\rho = ||\mathbf{W}^T\mathbf{W} - \mathbf{I}||, \tag{38}$$

where $||\cdot||$ refers to the Euclidean matrix norm. The columns of $\mathbf{W}$ are first normalized to unit length, so that ρ measures the deviation of the off-diagonal elements of $\mathbf{W}$ from zero. A smaller value of ρ indicates higher orthogonality and ρ equals to 0 when the columns of $\mathbf{W}$ are completely orthogonal.

Figure (3, top) compares the orthogonal behavior among PNMF, NNSC and NMF as the learning proceeds. PNMF converges to a local minimum with much lower ρ value, that is, higher orthogonality. Figure (3, bottom) shows that PNMF is not sensitive to the initial values.

6.3 MRI data

This data set consists of a single high-resolution anatomical volume obtained by Magnetic Resonance Imaging (MRI). The volume was acquired axially, with 90 horizontal slices parallel to the line connecting the anterior and posterior commissures. The source data matrix $\mathbf{V}$ has the size 65536×90, with the number of columns much less than the number of rows.

Running NMF and PNMF algorithms, the bases are shown in Figure 6.3 with the rank $r = 25$. Figure 6.3 shows the reconstructions of one of the images. It can be seen that PNMF is able to bring out considerably more details.

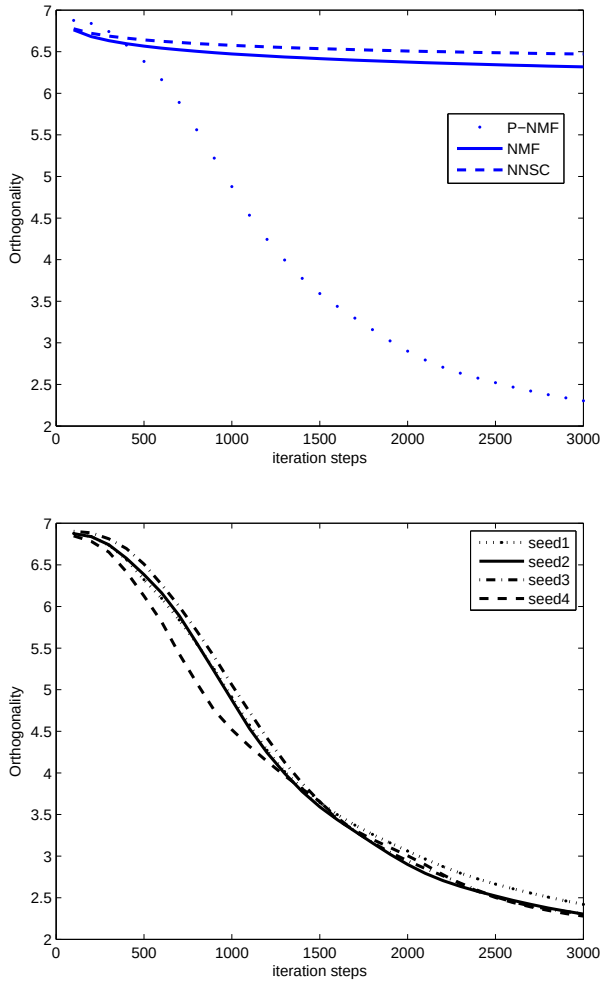

Fig. 3 Orthogonality versus iterative steps using NMF and PNMF with subdimension 49. Up: PNMF and NMF. Down: PNMF with four different random seeds.

7 Conclusion

We have 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, called Projective NMF (PNMF) is based on positively constrained projections and is related both to NMF and to the conventional SVD decomposition. Two iterative positive projection learning algorithms were suggested, one based on minimizing Euclidean distance and the other one on minimizing the divergence between 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 number of free parameters to be determined in PNMF

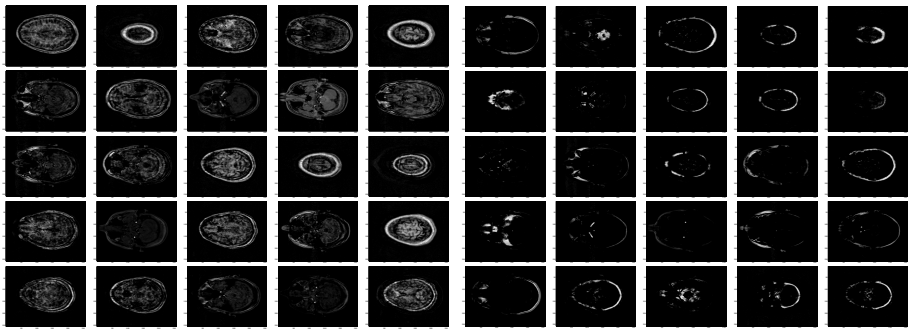

Fig. 4 NMF (left), PNMF method (right) bases of dimension 25. Each basis component consists of 256×256 pixels.

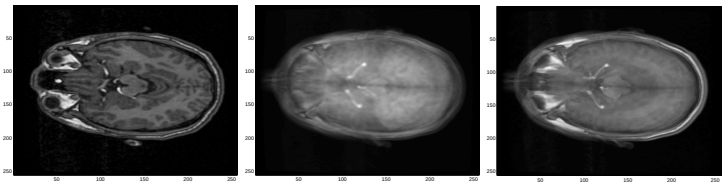

Fig. 5 The original image (left) and its reconstructions by NMF (middle) and PNMF under 100 iterative steps (right)

is much less than in NMF. The tradeoff is that the convergence, counted in iteration steps, is slower than in NMF.

One purpose of the non-negative factorization approaches is to learn localized features which would be suitable not only for image compression, but also for object recognition. Experimental results on face and biomedical images show that PNMF derives bases which are better suitable for a localized representation than NMF, with considerably more orthogonal basis vectors. The orthogonality has also the benefit that PNMF can be used for clustering in the same way as NMF. It was shown that the PNMF Euclidean cost function has a very close relation to k-means clustering, the difference being that PNMF produces a "soft" clustering in which the degree of belonging to a cluster is a continuous number instead of binary. The clustering result was experimentally shown to be somewhat better than for NMF.

References

1. Bell, A. and Sejnowski, T., The "independent components" of images are edge filters. *Vision Research*, 37: 3327–3338, 1997.
2. Cichocki, A., Morup, M., Smaragdis, P., Wang, W., and Zdunek, R., Advances in nonnegative matrix and tensor factorization. Computational Intelligence and Neuroscience, 2008: 852187. Published online July 2008.
3. Diamantaras, K. I. and Kung, S. Y., Principle Component Neural Networks: Theory and Applications. Wiley, 1996.
4. Ding, C., He, X. and Simon, D. H., On the Equivalence of Nonnegative Matrix Factorization and Spectral Clustering. Proc. SIAM Int'l Conf. Data Mining (SDM'05), pp. 606-610, April 2005
5. Ding, C., Li, T. and Peng, W., On the Equivalence Between Nonnegative Matrix Factorization and Probabilistic Latent Semantic Indexing. Computational Statistics and Data Analysis, 52: 3913-3927, 2008.

6. Golub, G. and Loan C. van, Matrix Computations. The Johns Hopkins University Press, 3rd edition, 1996.
7. Guillamet, D., Schiele, B. and Vitri, J., Analyzing non-negative matrix factorization for image classification. In Proc. 16th Internat. Conf. Pattern Recognition (ICPR02), Vol. II, 116119. IEEE Computer Society, August 2002.
8. Hoyer, P. O., Nonnegative sparse coding, Neural Networks for Signal Processing XII, Proc. IEEE Workshop on Neural Networks for Signal Processing, Martigny, 2002.
9. Hoyer, P. O., Non-negative Matrix Factorization with sparseness constraints. *Journal of Machine Learning Research*, 5:1457-1469, 2004.
10. Hyvärinen, A. and Hoyer, P. O., Emergence of phase and shift invariant features by decomposition of natural images into independent feature subspaces. *Neural Computation*, 13: 1527–1558, 2001.
11. Hyvärinen, A., Karhunen, J. and Oja, E., *Independent Component Analysis*. Wiley, New York, 2001.
12. Lee, D. D. and Seung, H. S., Learning the parts of objects by non-negative matrix factorization. *Nature*, 401:788–791, 1999.
13. Lee, D. D. and Seung, H. S., Algorithms for non-negative matrix factorization. In *NIPS*, pp. 556–562, 2000.
14. Li, T. and Ding, C., The Relationships Among Various Nonnegative Matrix Factorization Methods for Clustering. Proc. IEEE Int'l Conf. on Data Mining (ICDM'06), pp. 362-371, 2006.
15. Olshausen, B. A. and Field, D. J., Natural image statistics and efficient coding. *Network*, 7: 333–339, 1996.
16. Pauca, V., Shahnaz, F., Berry, M., Plemmons, R., Text Mining Using Non-Negative Matrix Factorizations. In: Proceedings of the Fourth SIAM International Conference on Data Mining, SIAM, Lake Buena Vista, FL. April 2004.
17. Paucaa, V. P., Pipera, J. and Plemmons, R. J., Nonnegative matrix factorization for spectral data analysis. Linear Algebra and its Applications, Vol. 416, Issue 1: 29-47, 2006.
18. Smaragdis, P., Non-negative matrix factor deconvolution; extraction of multiple sound sources from monophonic inputs. In 5th International Conference on Independent Component Analysis and Blind Source Separation (ICA04), pp. 494-499, Granada, Spain, 2004.
19. Smaragdis, P. and Brown, J. C., Non-negative matrix factorization for polyphonic music transcription. In IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA03), pp. 177-180, October 2003.
20. Hateren, J. H. van and Schaaf, A. van der, Independent component filters of natural images compared with simple cells in primary visual cortex. *Proc. Royal Soc. London B*, 265: 2315–2320, 1998.
21. Yang, Z., Yuan, Z. and Laaksonen, J., Projective Nonnegative Matrix Factorization with Applications to Facial Image Processing. International Journal of Pattern Recognition and Artificial Intelligence, 21(8):1353-1362, 2007.
22. Yuan, Z. and Oja, E., A family of Projective nonnegative matrix factorization algorithms. ISSPA 2007, Sharjah, United Arab Emirates.
23. Yuan, Z. and Oja, E., Projective nonnegative matrix factorization for image compression and feature extraction. In: Image Analysis Springer, Berlin, Germany, pp. 333-342, 2005.