Non-negative matrix factorization (NMF)

1. Introduction

General information on this model, notation, and references, can be found in the Appendix, which contains a published paper. Another excellent reference is the book:
A Cichocki, R Zdunek, A-H Phan, S Amari Nonnegative Matrix and Tensor Factorizations: Applications to Exploratory Multi-way Data Analysis and Blind Source Separation. John Wiley & Sons, Ltd, (2009)

The data must be strictly positive, no negative numbers, no zero values either.

Consider the NMF problem:

Eq. 1 $\mathbf{V} = \mathbf{WH}$

with $\mathbf{V} \in \mathbb{R}_{+}^{p \times N}$, $\mathbf{W} \in \mathbb{R}_{+}^{p \times q}$, $\mathbf{H} \in \mathbb{R}_{+}^{q \times N}$; p is number of variables, q is number of components, N is number of objects or items; and $q \leq rank(\mathbf{V})$. The columns of $\mathbf{W}$ sum to 1.

2. There are 2 basic algorithms.

2.a. Least squares with multiplicative updates

The minimization problem corresponds to the following metric:

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p}\sum_{j=1}^{N}\left(v_{ij} - \sum_{k=1}^{q} w_{ik}h_{kj}\right)^2 = \sum_{i=1}^{p}\sum_{j=1}^{N}\left(v_{ij}^2 + \left(\sum_{k=1}^{q} w_{ik}h_{kj}\right)^2 - 2v_{ij}\left(\sum_{k=1}^{q} w_{ik}h_{kj}\right)\right)$$

Focusing on w_{ab} gives:

$$D(w_{ab}) = \sum_{j=1}^{N}\left(\left(w_{ab}h_{bj} + \sum_{\substack{k=1 \\ k\neq b}}^{q} w_{ak}h_{kj}\right)^2 - 2v_{aj}\left(w_{ab}h_{bj} + \sum_{\substack{k=1 \\ k\neq b}}^{q} w_{ak}h_{kj}\right)\right)$$

with partial derivative:

$$\frac{\partial}{\partial w_{ab}} D(w_{ab}) = 2\sum_{j=1}^{N} h_{bj}\left(\sum_{k=1}^{q} w_{ak}h_{kj}\right) - 2\sum_{j=1}^{N} v_{aj}h_{bj}$$

which leads to:

$$w_{ab} = w_{ab} - \eta \frac{\partial}{\partial w_{ab}} D(w_{ab})$$

$$w_{ab} = w_{ab} + \eta\left[2\sum_{j=1}^{N} v_{aj}h_{bj} - 2\sum_{j=1}^{N} h_{bj}\left(\sum_{k=1}^{q} w_{ak}h_{kj}\right)\right]$$

and setting:

$$\eta = \frac{w_{ab}}{2\sum_{j=1}^{N} h_{bj}\left(\sum_{k=1}^{q} w_{ak}h_{kj}\right)}$$

gives:

$$w_{ab} = w_{ab} \frac{\sum_{j=1}^{N} v_{aj} h_{bj}}{\sum_{j=1}^{N} h_{bj} \left(\sum_{k=1}^{q} w_{ak} h_{kj} \right)}$$

If we denote:

$$\hat{v}_{ij} = \sum_{k=1}^{q} w_{ik} h_{kj}$$

then:

$$w_{ab} = w_{ab} \frac{\sum_{j=1}^{N} v_{aj} h_{bj}}{\sum_{j=1}^{N} \hat{v}_{aj} h_{bj}}$$

Similarly:

$$h_{bc} = h_{bc} \frac{\sum_{i=1}^{p} w_{ib} v_{ic}}{\sum_{i=1}^{p} w_{ib} \hat{v}_{ic}}$$

The algorithm iterates the last two multiplicative update equations. In detail:

1.a. Given data **V**; initial **W** and **H**, with columns of **W** sum to 1; and an exponent value r>0.

1.b. For all b and c: $h_{bc} = h_{bc} \frac{\sum_{i=1}^{p} w_{ib} v_{ic}}{\sum_{i=1}^{p} w_{ib} \hat{v}_{ic}}$

1.c. For all a and b: $w_{ab} = w_{ab} \frac{\sum_{j=1}^{N} v_{aj} h_{bj}}{\sum_{j=1}^{N} \hat{v}_{aj} h_{bj}}$

1.d. $w_{ik} \leftarrow w_{ik}^{r}$; $i = 1...p$, $k = 1...q$

1.e. Make columns of **W** sum to 1.

1.f. Go to 1.b. until convergence.

Note the effect of the sparseness exponent r:
If r>1, then **W** will be sparse, and **H** will be smooth (non-sparse).
If r<1, then **H** will be sparse, and **W** will be smooth (non-sparse).
If r=1, then **W** and **H** will be equally sparse.

This algorithm is also implemented for the case of a symmetric data matrix **V**.

2.b. Divergence metric (MultUpdates):

This corresponds to equations 16, 17, and 18, in the paper in the Appendix. There is one additional step, which corresponds to inserting the step "1.d." (the sparseness exponent *r*) from the previous algorithm, inserted after equation 17 and before equation 18 from the paper in the Appendix.

Note the effect of the sparseness exponent *r*:
If *r*>1, then **W** will be sparse, and **H** will be smooth (non-sparse).
If *r*<1, then **H** will be sparse, and **W** will be smooth (non-sparse).
If *r*=1, then **W** and **H** will be equally sparse.

This algorithm is also implemented for the case of a symmetric data matrix **V**.

3. About the data format and the NMF model

The data on file is in the form $\mathbf{V}^T \in \mathbb{R}_+^{N\times p}$. It is read in from the file, and then transposed to $\mathbf{V} \in \mathbb{R}_+^{p\times N}$, and the model as written in Eq. 1 is applied. On output, the program creates and writes files for $\mathbf{W}^T \in \mathbb{R}_+^{q\times p}$ and for $\mathbf{H}^T \in \mathbb{R}_+^{N\times q}$.

In practice, this means the following:

An sLORETA file is a matrix of 4-byte floats, written in PASCAL as:
array[1..NT,1..NV] of 4ByteFloat
where NT is number of time samples for time domain data, or number of frequencies for spectral data; and NV-6239, which is the number of voxels. Note that this is the data, and on file it is in the form of $\mathbf{V}^T \in \mathbb{R}_+^{N\times p}$. However, the model is applied to the transposed form, i.e. in Eq. 1, the data matrix $\mathbf{V} \in \mathbb{R}_+^{p\times N}$ corresponds to p=NV=6239 voxels; and N=NT number of time samples or frequencies. Therefore, the matrix $\mathbf{W} \in \mathbb{R}_+^{p\times q}$, contains in the columns sLORETA images (each one with p=NV=6239 voxels), and the matrix $\mathbf{H} \in \mathbb{R}_+^{q\times N}$ contains in the rows (each one with N=NT time or frequency samples) the time series of spectra for each of the sLORETA images.

The output files with $\mathbf{W}^T \in \mathbb{R}_+^{q\times p}$ and $\mathbf{H}^T \in \mathbb{R}_+^{N\times q}$ can be read into the loreta programs, where $\mathbf{W}^T \in \mathbb{R}_+^{q\times p}$ displays the q sloreta basis images, and $\mathbf{H}^T \in \mathbb{R}_+^{N\times q}$ displays their respective q time series or spectra.

4. Appendix

On next page

Nonsmooth Nonnegative Matrix Factorization (*ns*NMF)

Alberto Pascual-Montano, *Member*, *IEEE*, J.M. Carazo, *Senior Member*, *IEEE*, Kieko Kochi, Dietrich Lehmann, and Roberto D. Pascual-Marqui

Abstract—We propose a novel nonnegative matrix factorization model that aims at finding localized, part-based, representations of nonnegative multivariate data items. Unlike the classical nonnegative matrix factorization (NMF) technique, this new model, denoted "nonsmooth nonnegative matrix factorization" (*ns*NMF), corresponds to the optimization of an unambiguous cost function designed to explicitly represent sparseness, in the form of nonsmoothness, which is controlled by a single parameter. In general, this method produces a set of basis and encoding vectors that are not only capable of representing the original data, but they also extract highly localized patterns, which generally lend themselves to improved interpretability. The properties of this new method are illustrated with several data sets. Comparisons to previously published methods show that the new *ns*NMF method has some advantages in keeping faithfulness to the data in the achieving a high degree of sparseness for both the estimated basis and the encoding vectors and in better interpretability of the factors.

Index Terms—nonnegative matrix factorization, constrained optimization, datamining, mining methods and algorithms, pattern analysis, feature extraction or construction, sparse, structured, and very large systems.

1 Introduction

Nonnegative matrix factorization (NMF) [1], [2] has been introduced as a matrix factorization technique that produces a useful decomposition in the analysis of data. NMF decomposes the data as a product of two matrices that are constrained by having nonnegative elements. This method results in a reduced representation of the original data that can be seen either as a feature extraction or a dimensionality reduction technique. More importantly, NMF can be interpreted as a parts-based representation of the data due to the fact that only additive, not subtractive, combinations are allowed. This is possible because of the nonnegativity constraints imposed in this model, which, unlike other factorization methods, such as singular value decomposition (SVD) or independent component analysis (ICA) [3], are only capable of extracting holistic features from the data.

Formally, the nonnegative matrix decomposition can be described as follows:

$$\mathbf{V} \approx \mathbf{WH}, \tag{1}$$

where $\mathbf{V} \in \mathbb{R}^{p \times n}$ is a positive data matrix with p variables and n objects, $\mathbf{W} \in \mathbb{R}^{p \times q}$ are the reduced q basis vectors or factors, and $\mathbf{H} \in \mathbb{R}^{q \times n}$ contains the coefficients of the linear combinations of the basis vectors needed to reconstruct the original data (also known as encoding vectors). The main difference between NMF and other classical factorization models relies in the nonnegativity constraints imposed on both the basis ($\mathbf{W}$) and encoding vectors ($\mathbf{H}$). In this way, only additive combinations are possible:

$$(\mathbf{V})_{i\mu} \approx (\mathbf{WH})_{i\mu} = \sum_{a=1}^{q} W_{ia} H_{a\mu}. \tag{2}$$

NMF has been successfully used in diverse fields of science such as biomedical applications [4], [5], [6], [7], face and object recognition [8], [9], color science [10], [11], and polyphonic music transcription [12], among others. Increasing interest in this factorization technique is due to the intuitive nature of the method, which has the ability to extract additive parts of data sets that are highly interpretable, while reducing the dimensionality of the data at the same time.

Even if NMF has been presented and used as a method capable of finding the underlying parts-based structure of complex data, there is no explicit guarantee in the method to support this property other than the nonnegativity constraints. In fact, taking a closer look at the basis and encoding vectors produced by NMF, it is noticeable that there is a high degree of overlapping among basis vectors that contradict the intuitive nature of the "parts" [13]. In this sense, a matrix factorization technique capable of producing more localized, less overlapped feature representations of the data is highly desirable in many applications. In this direction, there are several reported attempts for solving this problem by making modifications to the original NMF functional to enforce sparseness on the basis vectors, the encoding vectors, or both [14], [15], [16], [17].

- *A. Pascual-Montano is with the Computer Architecture and System Engineering Department, Facultad de Ciencias Físicas, Universidad Complutense, 28040 Madrid. Spain. E-mail: pascual@fis.ucm.es.*
- *J.M. Carazo is with the National Center for Biotechnology-CSIC, Campus Universidad Autónoma de Madrid, 28049 Madrid. Spain. E-mail: carazo@cnb.uam.es.*
- *K. Kochi, D. Lehmann, and R.D. Pascual-Marqui are with the KEY Institute for Brain-Mind Research, University Hospital of Psychiatry, Lenggstr. 31, CH-8029 Zurich, Switzerland. E-mail: kieko@access.unizh.ch, {dlehmann, pascualm}@key.unizh.ch.*

Manuscript received 7 Apr. 2005; revised 26 July 2005; accepted 28 July 2005; published online 13 Jan. 2006.
Recommended for acceptance by R. Basri.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-0191-0405.

In this study, a different matrix factorization technique is presented, based on a new cost function and its optimization algorithm. The cost function is derived by introducing a modification to the Lee and Seung method [2] in order to demand sparseness to both the basis and encoding vectors. The new method, here referred to as Nonsmooth Nonnegative Matrix Factorization (*ns*NMF), differs from the original in the use of an extra smoothness matrix for imposing sparseness. The goal of *ns*NMF is to find sparse structures in the basis functions that explain the data set. The interpretation of the new factorization is twofold: Data can be faithfully reconstructed using additive combinations of the basis functions, while, at the same time, interpretation of the basis functions is straightforward. The underlying rationale is to find a positive decomposition of the data into nonoverlapping parts.

This paper is organized as follows: In Section 2, the classical NMF problem, its optimization algorithm, and a description of related methods are described. In Section 3, the original NMF functional is extended with a smoothing matrix to achieve sparseness on both the basis and encoding vectors. The optimization algorithm is also presented. Section 4 and 5 present some factorization examples on synthetic and real data sets. Finally, Section 6 contains some concluding remarks.

2 Review of Nonnegative Matrix Factorization (NMF) and Its Sparse Variants

In this section, we will briefly describe the original nonnegative matrix factorization method [2] and some of the works that, to the best of our knowledge, are closely related to the work presented here [14], [15], [16], [17].

2.1 Nonnegative Matrix Factorization (NMF)

A formal description of Nonnegative Matrix Factorization as described in [2] follows: Let:

$$\mathbf{V} \approx \mathbf{WH}, \tag{3}$$

where $\mathbf{V} \in \mathbb{R}^{p \times n}$ is a data matrix with p variables and n objects, $\mathbf{W} \in \mathbb{R}^{p \times q}$ are the factor vectors by columns, and $\mathbf{H} \in \mathbb{R}^{q \times n}$ contains the encoding vectors or projections, $q \leq p$, all matrices $\mathbf{V}, \mathbf{W}, \mathbf{H}$ are nonnegative, and the columns of $\mathbf{W}$ (the basis vectors) are normalized (sum up to 1).

The objective function, based on the Poisson likelihood, is:

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p} \sum_{j=1}^{n} \left(V_{ij} \ln \frac{V_{ij}}{(\mathbf{WH})_{ij}} - V_{ij} + (\mathbf{WH})_{ij} \right), \tag{4}$$

which, after some simplifications and elimination of pure data terms, gives:

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p} \sum_{j=1}^{n} \left(\sum_{k=1}^{q} W_{ik} H_{kj} - V_{ij} \ln \sum_{k=1}^{q} W_{ik} H_{kj} \right). \tag{5}$$

Taking the derivative with respect to $\mathbf{H}$ gives:

$$\frac{\partial}{\partial H_{ab}} D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p} W_{ia} - \sum_{i=1}^{p} \frac{V_{ib} W_{ia}}{\sum_{k=1}^{q} W_{ik} H_{kb}}. \tag{6}$$

The gradient algorithm then states:

$$H_{ab} \leftarrow H_{ab} - \eta_{ab} \frac{\partial}{\partial H_{ab}} D(\mathbf{V}, \mathbf{WH}), \tag{7}$$

$$H_{ab} \leftarrow H_{ab} + \eta_{ab} \left[\sum_{i=1}^{p} \frac{V_{ib} W_{ia}}{\sum_{k=1}^{q} W_{ik} H_{kb}} - \sum_{i=1}^{p} W_{ia} \right], \tag{8}$$

for some step size η_{ab}.

Forcing:

$$\eta_{ab} = \frac{H_{ab}}{\sum_{i=1}^{p} W_{ia}} \tag{9}$$

gives the multiplicative rule:

$$H_{ab} \leftarrow H_{ab} \frac{\sum_{i=1}^{p} (W_{ia} V_{ib}) / \sum_{k=1}^{q} W_{ik} H_{kb}}{\sum_{i=1}^{p} W_{ia}}. \tag{10}$$

Taking the derivative with respect to $\mathbf{W}$ gives:

$$\frac{\partial}{\partial W_{cd}} D(\mathbf{V}, \mathbf{WH}) = \sum_{j=1}^{n} H_{dj} - \sum_{j=1}^{n} \frac{V_{cj} H_{dj}}{\sum_{k=1}^{q} W_{ck} H_{kj}}. \tag{11}$$

The gradient algorithm then states:

$$W_{cd} \leftarrow W_{cd} - \nu_{cd} \frac{\partial}{\partial W_{cd}} D(\mathbf{V}, \mathbf{WH}), \tag{12}$$

$$W_{cd} \leftarrow W_{cd} + \nu_{cd} \left[\sum_{j=1}^{n} V_{cj} \frac{H_{dj}}{\sum_{k=1}^{q} W_{ck} H_{kj}} - \sum_{j=1}^{n} H_{dj} \right]. \tag{13}$$

Forcing the step size:

$$\nu_{cd} = \frac{W_{cd}}{\sum_{j=1}^{n} H_{dj}} \tag{14}$$

gives:

$$W_{cd} \leftarrow W_{cd} \frac{\sum_{j=1}^{n} (H_{dj} V_{cj}) / \sum_{k=1}^{q} W_{ck} H_{kj}}{\sum_{j=1}^{n} H_{dj}}. \tag{15}$$

Finally, the derived algorithm is as follows:

1. Initialize $\mathbf{W}$ and $\mathbf{H}$ with positive random numbers.
2. For each basis vector $\mathbf{W}_a \in \mathbb{R}^{p\times 1}$, update the corresponding encoding vector $\mathbf{H}_a \in \mathbb{R}^{1\times n}$, followed by updating and normalizing the basis vector $\mathbf{W}_a$. Repeat this process until convergence.

Formally, the detailed algorithm follows:

Repeat until convergence:
For a = 1...q do begin
For b = 1...n do

$$H_{ab} \leftarrow H_{ab} \frac{\sum_{i=1}^{p}(W_{ia}V_{ib}) / \sum_{k=1}^{q} W_{ik}H_{kb}}{\sum_{i=1}^{p} W_{ia}}. \tag{16}$$

For c=1...p do begin

$$W_{ca} \leftarrow W_{ca} \frac{\sum_{j=1}^{n}\left(H_{aj}V_{cj}\right) / \sum_{k=1}^{q} W_{ck}H_{kj}}{\sum_{j=1}^{n} H_{aj}}. \tag{17}$$

$$W_{ca} \leftarrow \frac{W_{ca}}{\sum_{j=1}^{n} W_{ja}}. \tag{18}$$

End

End

2.2 Local Nonnegative Matrix Factorization (LNMF)

Inspired by the original NMF method [2], Feng et al. [14] introduced the Local Nonnegative Matrix Factorization (LNMF) algorithm intended for learning spatially localized, parts-based representation of visual patterns. Their aim was to obtain a truly part-based representation of objects by imposing sparseness constraints on the encoding vectors (matrix $\mathbf{H}$) and locality constraints to the basis components (matrix $\mathbf{W}$). Those constraints, in addition to nonnegativity, produced an algorithm able to extract binary-like, quasi-orthogonal basis components.

Formally, the problem was defined as follows:

Taking the factorization problem defined in (1), define $\mathbf{A} = \left[a_{ij}\right] = \mathbf{W}^t\mathbf{W}$ and $\mathbf{B} = \left[b_{ij}\right] = \mathbf{H}\mathbf{H}^t$, where $\mathbf{A}, \mathbf{B} \in \mathbb{R}^{q\times q}$. The LNMF algorithm is based on the following three additional constraints:

1. *Maximum Sparseness in* $\mathbf{H}$. It should contain as many zero components as possible. This implies that the number of basis components required to represent $\mathbf{V}$ is minimized. Mathematically, each a_{ij} should be minimum.
2. *Maximum expressiveness of* $\mathbf{W}$. This constraint is closely related to the previous one and it aims at further enforcing maximum sparseness in $\mathbf{H}$. Mathematically, $\sum_{i=1}^{q} b_{ii}$ should be maximum.
3. *Maximum orthogonality of* $\mathbf{W}$. This constraint imposes that different bases should be as orthogonal as possible to minimize redundancy. This is forced by minimizing $\sum_{\forall i,j,i\neq j} a_{ij}$. Combining this constraint, with the one described in point 1, the objective is to minimize $\sum_{\forall i,j} a_{ij}$.

Thus, $\forall i, j$, the constrained divergence function described in [14], is:

$$\begin{aligned} &D(\mathbf{V}, \mathbf{WH}) = \\ &\sum_{i=1}^{p}\sum_{j=1}^{n}\left(V_{ij}\ln\frac{V_{ij}}{(\mathbf{WH})_{ij}} - V_{ij} + (\mathbf{WH})_{ij}\right) \\ &+ \alpha\sum_{i,j=1}^{q} a_{ij} - \beta\sum_{i=1}^{q} b_{ii}, \end{aligned} \tag{19}$$

where $\alpha, \beta > 0$ represent some constants for expressing the importance of the additional constraints described above. One possible solution for this problem was described in [14] and is very similar to the original NMF algorithm described in [2]. Basically, it consists of the following steps:

Repeat until convergence:
For a =1...q do begin
For b = 1...n do

$$H_{ab} \leftarrow \sqrt{H_{ab}\sum_{i=1}^{p}(W_{ia}V_{ib}) / \sum_{k=1}^{q} W_{ik}H_{kb}}. \tag{20}$$

For c = 1...p
Update $\mathbf{W}$ using (17) and (18)
End

Notice that the minimization algorithm described in [14] eliminated the use of α and β. Even if this might look like a practical advantage, it also limits control over the sparseness constraints.

2.3 Nonnegative Sparse Coding (NNSC)

Similar to the LNMF algorithm, the Nonnegative Sparse Coding (NNSC) [15] method is intended to decompose multivariate data into a set of positive sparse components by using theory inherited from Linear Sparse Coding [18], [19]. Combining a small reconstruction error with a sparseness criterion, the objective function defined in [15] is:

$$E(\mathbf{V}, \mathbf{WH}) = \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|^2 + \lambda\sum_{i=1}^{q}\sum_{j=1}^{n} f\left(H_{ij}\right), \tag{21}$$

where the form of f defines how sparseness on $\mathbf{H}$ is measured and λ controls the trade-off between sparseness and the accuracy of the reconstruction. In [15], the authors used a linear activation penalty function to measure the sparseness, leading to the following objective function and its minimization algorithm:

$$E(\mathbf{V}, \mathbf{WH}) = \frac{1}{2}\|\mathbf{V} - \mathbf{WH}\|^2 + \lambda\sum_{i=1}^{q}\sum_{j=1}^{n} H_{ij}. \tag{22}$$

Algorithm

1. Initialize $\mathbf{W}$ and $\mathbf{H}$ to random strictly positive matrices of the appropriate dimensions, and normalize each column of $\mathbf{W}$. Let $\mu > 0$ denote the step-size.
2. Iterate until convergence:
   a. Calculate new $\mathbf{W}$as
   $$\mathbf{W} \leftarrow \mathbf{W} - \mu(\mathbf{WH} - \mathbf{V})\mathbf{H}^t. \tag{23}$$
   b. Any negative values in $\mathbf{W}$ are set to zero
   c. Normalize each column of $\mathbf{W}$.
   d. Calculate new $\mathbf{H}$ as
   $$H_{i,j} \leftarrow H_{i,j}\frac{(\mathbf{W}^t\mathbf{V})_{ij}}{(\mathbf{W}^t\mathbf{WH})_{ij} + \lambda)}. \tag{24}$$

2.4 Sparse Nonnegative Matrix Factorization (SNMF)

Liu et al. [17] modified the method described previously [15]. Instead of using a Euclidean least-square type functional, as in (22), they used a divergence term as in the original NMF paper [2] (see, e.g., (4)). Thus, the sparse NMF functional is:

$$D(\mathbf{V}, \mathbf{WH}) = \sum_{i=1}^{p}\sum_{j=1}^{n}\left(V_{ij}\ln\frac{V_{ij}}{(\mathbf{WH})_{ij}} - V_{ij} + (\mathbf{WH})_{ij}\right) + \alpha\sum_{ij}H_{ij} \tag{25}$$

for $\alpha \geq 0$.

This method forces sparseness via minimizing the sum of all H_{ij}. The update rule for matrix $\mathbf{H}$ is:

$$H_{ab} \leftarrow H_{ab}\frac{\sum_{i=1}^{p}(W_{ia}V_{ib}) / \sum_{k=1}^{q}W_{ik}H_{kb}}{1+\alpha}, \tag{26}$$

while the update rule for $\mathbf{W}$ is expressed in (17) and (18).

2.5 Nonnegative Matrix Factorization with Sparseness Constraints (NMFSC)

A more recent work related to the addition of sparseness constraints to the classical NMF problem has also been proposed by Hoyer [16]. This method minimizes $E(\mathbf{V}, \mathbf{WH}) = \|\mathbf{V} - \mathbf{WH}\|^2$ under the following constraints:

$$\text{Sparseness}(\mathbf{W}_i) = S_w, \forall_i, i = 1..q,$$
$$\text{Sparseness}(\mathbf{H}_i) = S_h, \forall_i, i = 1..q,$$

where $\mathbf{W}_i$ is the ith column of $\mathbf{W}$, $\mathbf{H}_i$ is the ith row of $\mathbf{H}$, S_w and S_h are the desired sparseness values for $\mathbf{W}$ and $\mathbf{H}$, respectively, and are user-defined parameters. The sparseness criteria proposed in [16] uses a measure based on the relationship between the L_1 and L_2 norm of a given vector:

$$\text{Sparseness}(\mathbf{x}) = \frac{\sqrt{n} - \left(\sum|x_j|\right)/\sqrt{\sum x_j^2}}{\sqrt{n-1}}, \tag{27}$$

where n is the dimensionality of the vector $\mathbf{x}$.

This sparseness measure quantifies how much energy of a vector is packed into only a few components. This function evaluates to 1 if and only if $\mathbf{x}$ contains only a single nonzero component, and takes a value of 0 if and only if all components are equal, interpolating smoothly between the two extremes.

Details of the lengthy algorithm are omitted here and can be found in [16].

3 Our Proposal: Nonsmooth Nonnegative Matrix Factorization (*ns*NMF)

All the methods described in the previous section try to achieve further sparseness in the nonnegative matrix factorization model by means of the ad hoc addition of constraints or penalization terms to the divergence functional or to the Euclidean least squares functional. Such constraints or penalizations can be applied to the basis vectors alone, to the encoding vectors alone, or simultaneously to both basis and encoding vectors.

Because of the multiplicative nature of the model, i.e., "basis" multiplied by "encoding," sparseness in one of the factors will almost certainly force "nonsparseness" or smoothness in the other, in order to compensate for the final product to reproduce the data as best as possible. On the other hand, forcing sparseness constraints on both the basis and the encoding vectors will deteriorate the goodness of fit of the model to the data. Therefore, from the outset, this approach is doomed to failure in achieving generalized sparseness and satisfactory goodness of fit.

These critical aspects of the previously published methods have motivated the direct modification of the model as the means to achieve global sparseness. The new model proposed in this study, denoted as "NonSmooth Nonnegative Matrix Factorization" (*ns*NMF), is defined as:

$$\mathbf{V} = \mathbf{WSH}, \tag{28}$$

where $\mathbf{V}$, $\mathbf{W}$, and $\mathbf{H}$ are the same as in the original NMF model. The positive symmetric matrix $\mathbf{S} \in \mathbb{R}^{q\times q}$ is a "smoothing" matrix defined as:

$$\mathbf{S} = (1-\theta)\mathbf{I} + \frac{\theta}{q}\mathbf{1}\mathbf{1}^T, \tag{29}$$

where $\mathbf{I}$ is the identity matrix, $\mathbf{1}$ is a vector of ones, and the parameter θ satisfies $0 \leq \theta \leq 1$.

The interpretation of $\mathbf{S}$ as a smoothing matrix can be explained as follows: Let $\mathbf{X}$ be a positive, nonzero, vector. Consider the transformed vector $\mathbf{Y} = \mathbf{SX}$. If $\theta = 0$, then $\mathbf{Y} = \mathbf{X}$ and no smoothing on $\mathbf{X}$ has occurred. However, as $\theta \to 1$, the vector $\mathbf{Y}$ tends to the constant vector with all elements almost equal to the average of the elements of $\mathbf{X}$. This is the smoothest possible vector in the sense of "nonsparseness" because all entries are equal to the same nonzero value, instead of having some values close to zero and others clearly nonzero.

Note that the parameter θ controls the extent of smoothness of the matrix operator $\mathbf{S}$. However, due to the multiplicative nature of the model (28), strong smoothing in $\mathbf{S}$ will force strong sparseness in both the basis and the encoding vectors in order to maintain faithfulness of the model to the data. Therefore, the parameter θ controls the sparseness of the model. Note that, when $\theta = 0$, the model corresponds to the basic NMF.

Further insight into the nature of the new *ns*NMF model can be obtained from the dual interpretation of (28), which can be equivalently written as:

TABLE 1
Comparison of Results for Different NMF Models for Different Levels of Sparseness

Method	Sparseness constraint	Explained variance (%)	Average sparseness on W	Average sparseness on H
NMF	-	99.99	0.64	0.20
LNMF	-	93.13	0.88	0.05
SNMF	0.1	99.17	0.63	0.20
	0.2	97.22	0.62	0.20
	0.3	94.67	0.67	0.18
	0.4	91.84	0.63	0.20
	0.5	88.89	0.65	0.19
	0.6	85.93	0.61	0.21
	0.7	83.04	0.64	0.19
	0.8	80.25	0.63	0.19
	0.9	77.43	0.59	0.22
NMFSC	0.1	87.11	0.1	0.1
	0.2	93.54	0.2	0.2
	0.3	97.7	0.3	0.3
	0.4	99.30	0.4	0.4
	0.5	96.67	0.5	0.5
	0.6	84.59	0.6	0.6
	0.7	64.75	0.7	0.7
	0.8	38.83	0.8	0.8
	0.9	26.18	0.9	0.9
*ns*NMF	0.1	99.99	0.66	0.21
	0.2	99.99	0.71	0.22
	0.3	99.99	0.78	0.21
	0.4	99.98	0.84	0.22
	0.5	98.92	0.54	0.45
	0.6	99.30	0.87	0.26
	0.7	98.36	0.87	0.29
	0.8	95.65	0.56	0.52
	0.9	94.24	0.86	0.42

$$\mathbf{V} = (\mathbf{WS})\mathbf{H} = \mathbf{W}(\mathbf{SH}).$$

Nonsparseness in the basis $\mathbf{W}$ will force sparseness in the encoding $\mathbf{H}$. At the same time, nonsparseness in the encoding $\mathbf{H}$ will force sparseness in the basis $\mathbf{W}$. Due precisely the simultaneity of both conditions, sparseness will be enforced on both basis and encoding parts.

The new algorithm is very straightforward to derive by simply substituting the *ns*NMF model (28) into the divergence functional in (4) and following the same procedure to minimize the functional as performed in (5)-(15). For a given sparseness parameter value $0 \leq \theta \leq 1$, the final algorithm is a simple modification of the original, basic NMF algorithm given by (16)-(18):

1. In the update equation for $\mathbf{H}$ (16), substitute $(\mathbf{W})$ with $(\mathbf{WS})$.
2. In the update equation for $\mathbf{W}$ (17), substitute $(\mathbf{H})$ with $(\mathbf{SH})$.
3. Equation (18) remains the same.

4 Experiments

4.1 Synthetic Data Set

As mentioned in the previous section, the multiplicative nature of the sparse variants of the NMF model will produce a paradoxical effect: Imposing sparseness in one of the factors will almost certainly force smoothness in the other in an attempt to reproduce the data as best as possible. Additionally, forcing sparseness constraints on both the basis and the encoding vectors will decrease the explained variance of the data by the model. To demonstrate that our new *ns*NMF method is less susceptible to this effect, we carried out a simple experiment with an artificial data set composed of five variables measured on 20 objects (items). This was generated by multiplying two matrices, $\mathbf{A} \in \mathbb{R}^{5\times 3}$ and $\mathbf{B} \in \mathbb{R}^{3\times 20}$, where each element of each matrix was assigned an independent uniform [0, 1] pseudorandom number. There was no sparseness included into this data set.

The only feature in this data set is that it is positive and that it can be represented exactly with only three factors (three basis and three encoding vectors). This property is ideal for studying the deterioration of goodness of fit between the data and the model, as a function of sparseness, for the different methods.

Table 1 shows the results when using exactly three factors in all cases. Different NMF-type methods were applied to the same randomly generated positive data set (5 variables, 20 items, rank = 3). The methods are denoted as NMF, LNMF, SNMF, NMFCS, and the new *ns*NMF method, as described in Section 2. When applicable, different levels of sparseness were used. The table reports the explained variance achieved by the factorization as well as the average sparseness of the basis vectors and of the encoding vectors (see (27)). Note that "average sparseness" is defined as the average of the

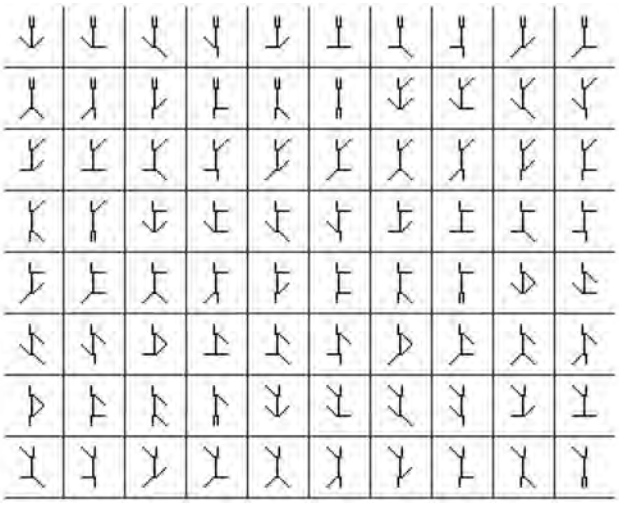

Fig. 1. Sample images from the *swimmer* data set. Each image is composed of a torso and four limbs in different positions.

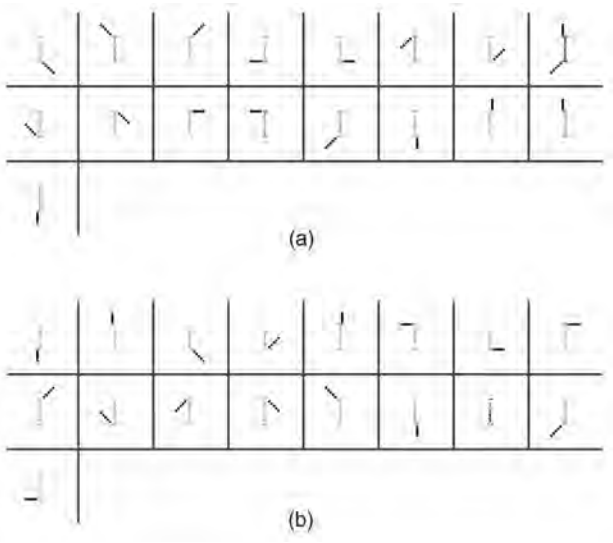

Fig. 2. Results of applying (a) NMF and (b) *ns*NMF algorithms to the *swimmer* data set. In both cases, 17 factors were generated. A value of $\theta = 0.5$ was used for the *ns*NMF model. Note that, in (a), the eighth factor of the first row represents a nonsparse representation of the two limbs and a torso, while, in (b), the seventh factor in the second row contains a stronger torso signal.

sparseness value (given by (27)) over all factors (columns of matrix $\mathbf{W}$) and encoding vectors (rows of matrix $\mathbf{H}$).

As expected, NMF achieves 100 percent explained variance, with low sparseness values (using (21)). The LNMF method, which has no control over the extent of sparseness, explains only 93 percent of the variance while achieving high sparseness for the basis vectors, but extremely low sparseness for the encoding vectors.

Despite the fact that the SNMF method is designed to control sparseness, it seems to be incapable of obtaining an actual increase in sparseness, while the explained variance deteriorates tremendously. The NMFSC method performs as expected, enforcing sparseness, but at the expense of a dramatic loss of faithfulness between the data and the model. Finally, the new *ns*NMF model maintains almost perfect faithfulness to the data (> 99.9 percent explained

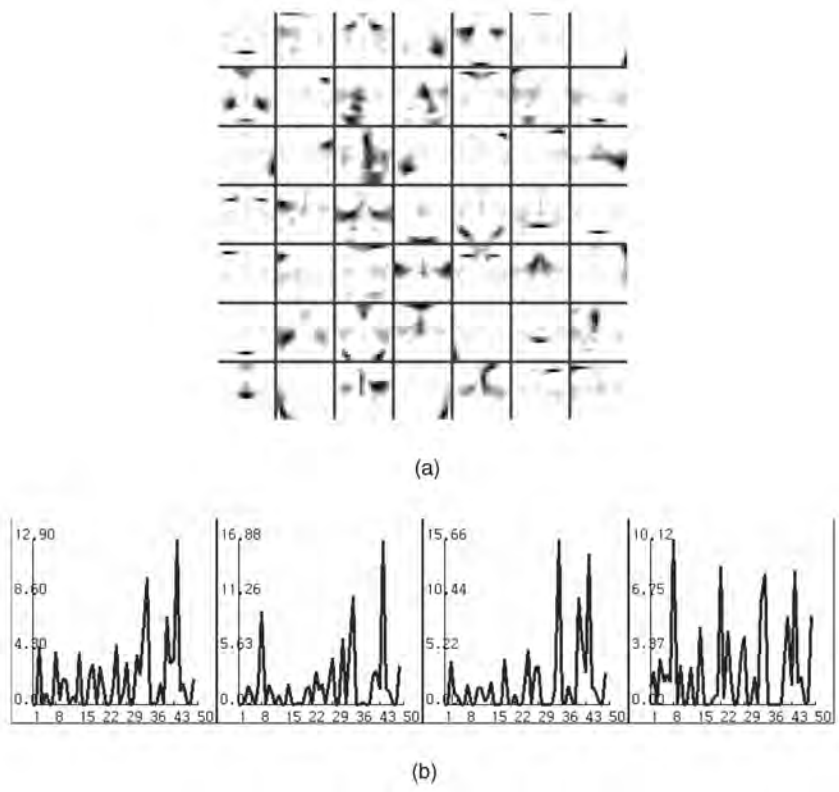

Fig. 3. Results of the NMF algorithm applied to the faces data set. (a) Forty-nine NMF basis components representing average parts of the faces contained in the database. (b) Coefficients (encodings) of four sample images. Total explained variance of the model: 95.72 percent.

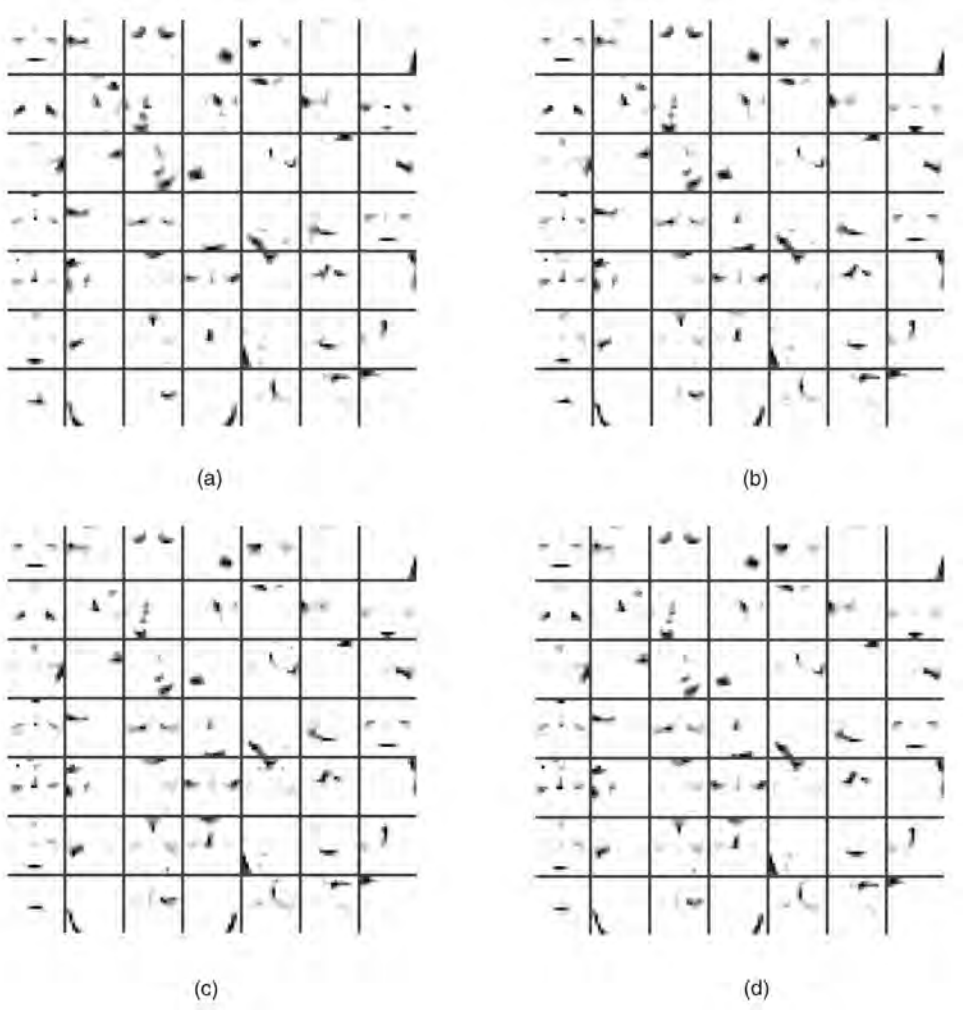

Fig. 4. *ns*NMF basis components with facial data set for different values of θ. (a) $\theta = 0.5$. Total explained variance of the model: 83.84 percent. (b) $\theta = 0.6$. Total explained variance of the model: 80.69 percent. (c) $\theta = 0.7$. Total explained variance of the model: 78.17 percent. (d) $\theta = 0.8$. Total explained variance of the model: 76.44 percent.

variance) for a wide range of achieved sparseness, thus outperforming the other methods.

4.2 Swimmers Data Set

The "*swimmer*" data set is described in [20]. It consists of a set of black-and-white images with four moving parts (limbs), each able to exhibit four different positions (articulations). Each individual image contains a "torso" of 12 pixels in the center and four "limbs" of six pixels that can be in one of four different positions. In total, there are 256 images of dimension 32×32 pixels, containing all possible limb positions/combinations. Fig. 1 shows a subset of these images.

The swimmer data set was used in [20] to demonstrate the ability of the NMF algorithm to find the parts (limbs). It was specifically created by an NMF-style generative model obeying some predefined rules, such as separability and complete factorial sampling. In [20], NMF was tested with this data set to demonstrate its capacity in finding parts. To that end, 16 factors were used. The factors showed that the 16 different articulated parts were properly resolved and perfectly agreed with the list of generators (four limbs in four different positions and one common torso) [20].

However, it is worth emphasizing that, in that study, the torso is not properly resolved since it is explicitly an invariant region that violated the rules used for generating the data. With the purpose of testing the "interpretability" of the factorizations, we reanalyzed this data using 17 factors instead of only 16. The idea was to check the ability of NMF and *ns*NMF in finding the limbs and the torso separately, as is expected in a parts-based representation of this data. Fig. 2 shows the best results (according to the functional values) of the NMF and *ns*NMF methods selected from 20 independent runs, each with random initializations. It can be noticed from Fig. 2 that NMF failed in extracting the 16 limbs and the torso, while *ns*NMF successfully explained the data using one factor for each independent part. These results are in total agreement with the nature of both methods: NMF extract parts of the data in a more holistic manner, while *ns*NMF sparsely represents the same reality.

4.3 Faces Data Set

In order to test the sparseness ability of the proposed algorithm, we applied *ns*NMF to the CBCL face database from MIT [21]. The database contains 2,429 19×19 facial low resolution gray-level images. The same data set has also been used in [2] to present the capacity of the NMF algorithm to produce a part-based representation of the

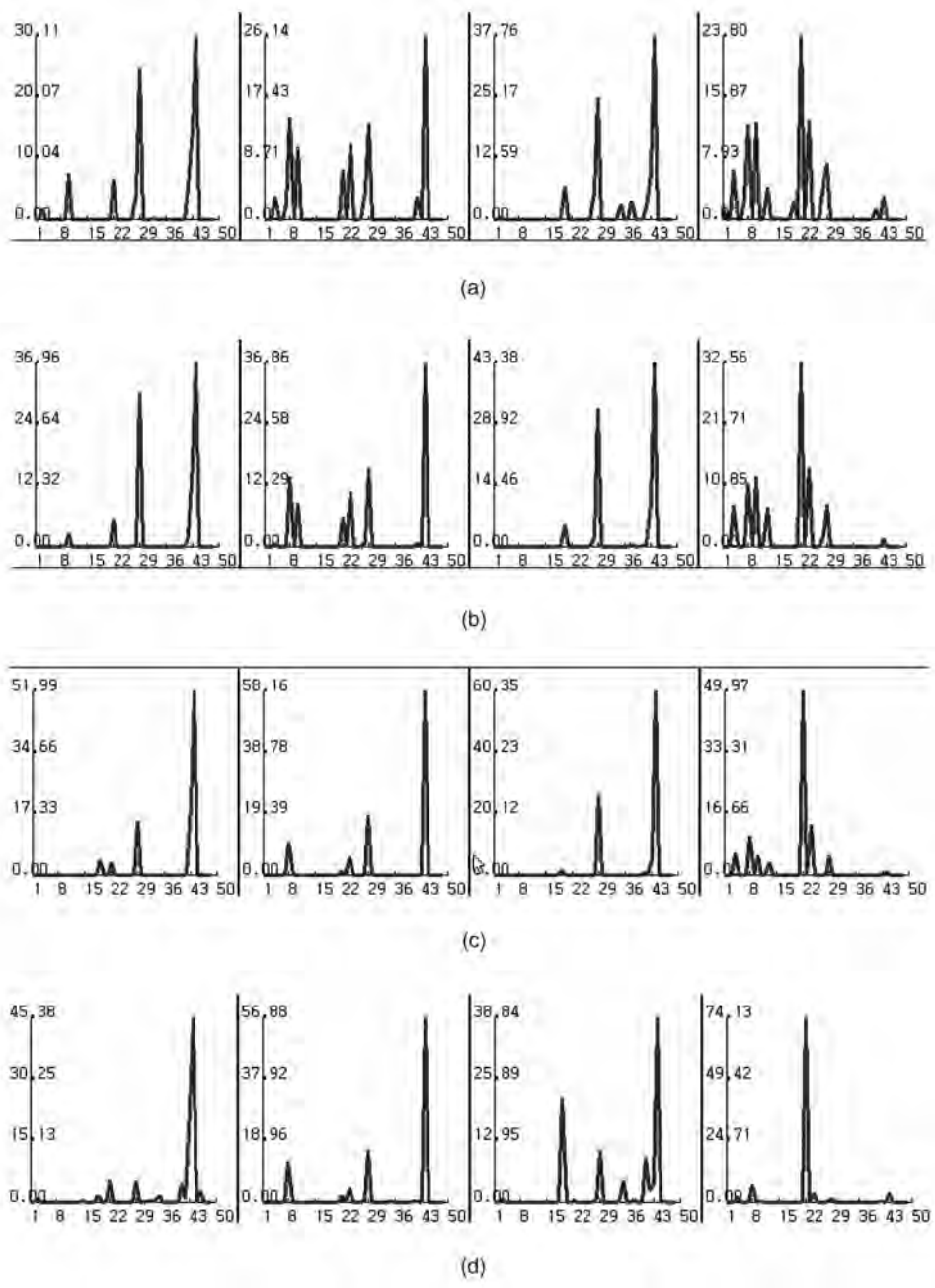

Fig. 5. Encoding vectors for four sample images and different values of sparseness: (a) $\theta = 0.5$, (b) $\theta = 0.6$, (c) $\theta = 0.7$, (d) $\theta = 0.8$. The effect of sparseness in the encoding vectors is more evident with the increase of θ.

facial images. For comparison purposes, the same number of factors (49) has been used in our experiments.

4.3.1 NMF Results

Fig. 3 shows the results using the Lee and Seung algorithm [2] applied to the facial database using 49 factors. Even if the factors' images give an intuitive notion of a parts-based representation of the original faces, the factorization is not really sparse enough to represent unique parts of an average face. In other words, the NMF algorithm allows some undesirable overlapping of parts, especially in those areas that are common to most of the faces in the input data. Fig. 3b shows the encoding coefficients for four sample images. As can be noticed, the coefficients are not sparse. This corroborates the hypothesis that, in order to reproduce the input images, the generated model needs to combine almost all of the factors in different overlapped proportions.

4.3.2 nsNMF Results

The *ns*NMF algorithm was applied to the facial database with different set of sparseness parameters $\theta = 0.5$, $\theta = 0.6$, $\theta = 0.7$, and $\theta = 0.8$. Figs. 4 and 5 show the results.

These results demonstrate the intrinsic nature of the *ns*NMF algorithm where more localized features appear with increasing values of the sparseness parameters. Since sparseness is equally applied on both the bases images and the coefficients, the sparse features are also observed in both. This result has a beneficial practical implication due to the fact that

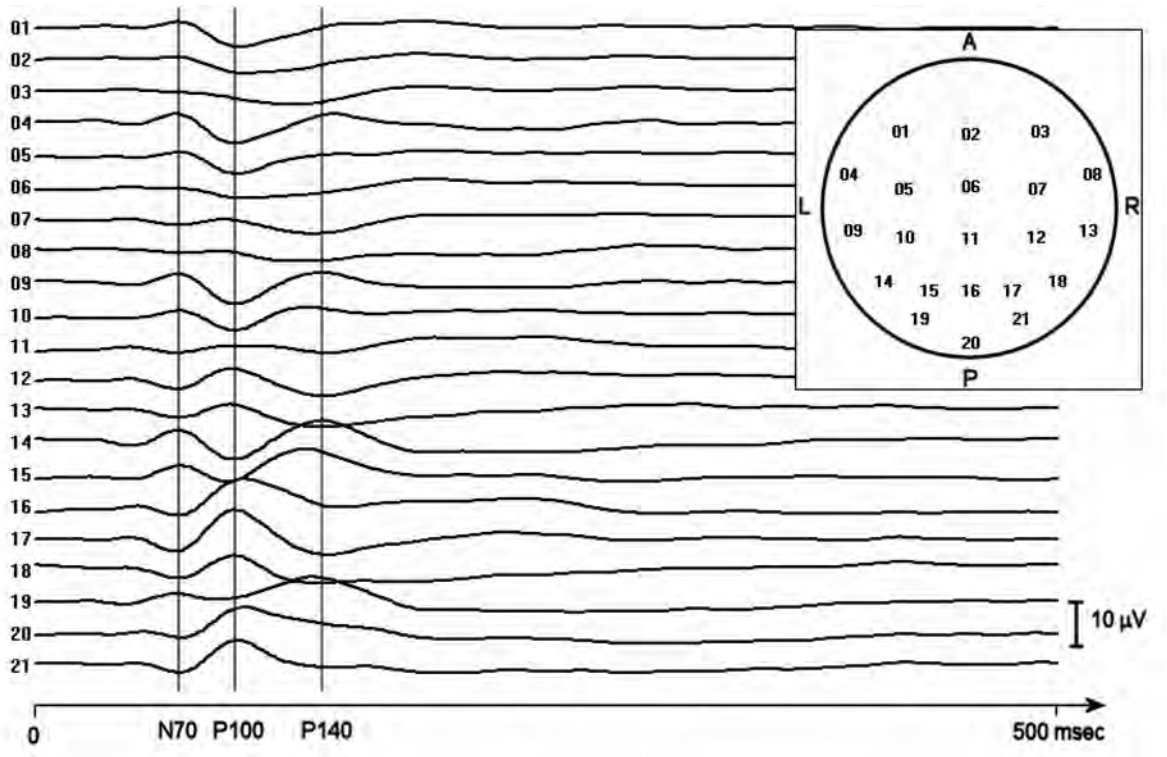

Fig. 6. Grand average scalp electric potentials (21 electrodes) due to right visual field stimulation with pattern reversal checkerboard. The upper-right inset illustrates the distribution of the electrodes over the scalp, as seen from above, A = anterior, P = posterior, L = left ear, R = right ear. For the vertical voltage axis, upward is positive and downward is negative. The three vertical cursors correspond to the main components (peaks) of the brain response in this experiment and they are denoted as N70 (70 msec), P100 (100 msec), and P140 (140 msec).

this new algorithm is not only able to extract localized features from a given data set, but also because it tries to explain each item in the data set by the additive combination of a minimum number of components (Fig. 5). This allows the recognition of the most important parts for a given item and, thus, it facilitates the interpretation stage in the data analysis process, as will be show, in the next section.

The faces data set has also been used in most of the previously related works [14], [15], [16], [17] to graphically illustrate the sparseness properties of their methods. Interested readers can also find a visual representation of the sparse faces decomposition in those related contributions to make a more visual qualitative comparison with the method presented here.

5 Real Application: Low-Resolution Brain Electromagnetic Tomography (LORETA)

This data set consists of time series of electric neuronal activity (current density) calculated at a large number of voxels located over the human cortex. This is an enormous amount of spatio-temporal data, thus making the task of understanding the mechanisms of brain information processing very difficult. One approach to aid in interpreting such complicated data is to model the spatio-temporal current densities in terms of a small number of factor pairs. In this model, the basis vectors would be certain normalized spatial distributions that have maximum activity in brain areas that are specialized in certain cognitive functions and the encoding vectors would be the time course of activation of the corresponding spatial distribution basis vector. However, this type of model will be simple only if the representations are sparse: The basis vectors consisting of the normalized spatial distribution of current density should be zero almost everywhere, except for nonzero values at some few brain regions, and the encoding vectors consisting of time course of activation should be zero almost everywhere, except at certain time intervals.

The original experimental data, known as an event related potential (ERP), corresponds to time varying measurements of scalp electric potential differences obtained while a human subject is viewing, in the right visual field, a computer screen displaying a checkerboard pattern that is reversing 1.1 times per second. Potentials were recorded at a sampling rate of 512 Hz from 21 scalp electrodes. The average brain response, time locked to pattern reversal onset, is known as the average ERP. Seventy pattern reversals per subject were averaged. This experiment was performed on 21 subjects and the grand average ERP over all subjects was analyzed. These data are described in further detail in [22].

In general, scalp electric potentials are due to electrically active neurons distributed over the cortex. The estimation of the spatial distribution of electric neuronal activity based on scalp potentials is known as a solution to the inverse problem of electroencephalography. In this study, we employed the low-resolution electromagnetic tomography (LORETA) method [22]. This tomography has been theoretically and experimentally validated [23], [24]. In its current implementation, it computes electric neuronal activity at 2,394 voxels distributed over the human cortex, using a standardized head model [25].

Fig. 6 illustrates the grand-average ERP for 21 electrodes and 256 discrete time samples (total time 0.5 sec).

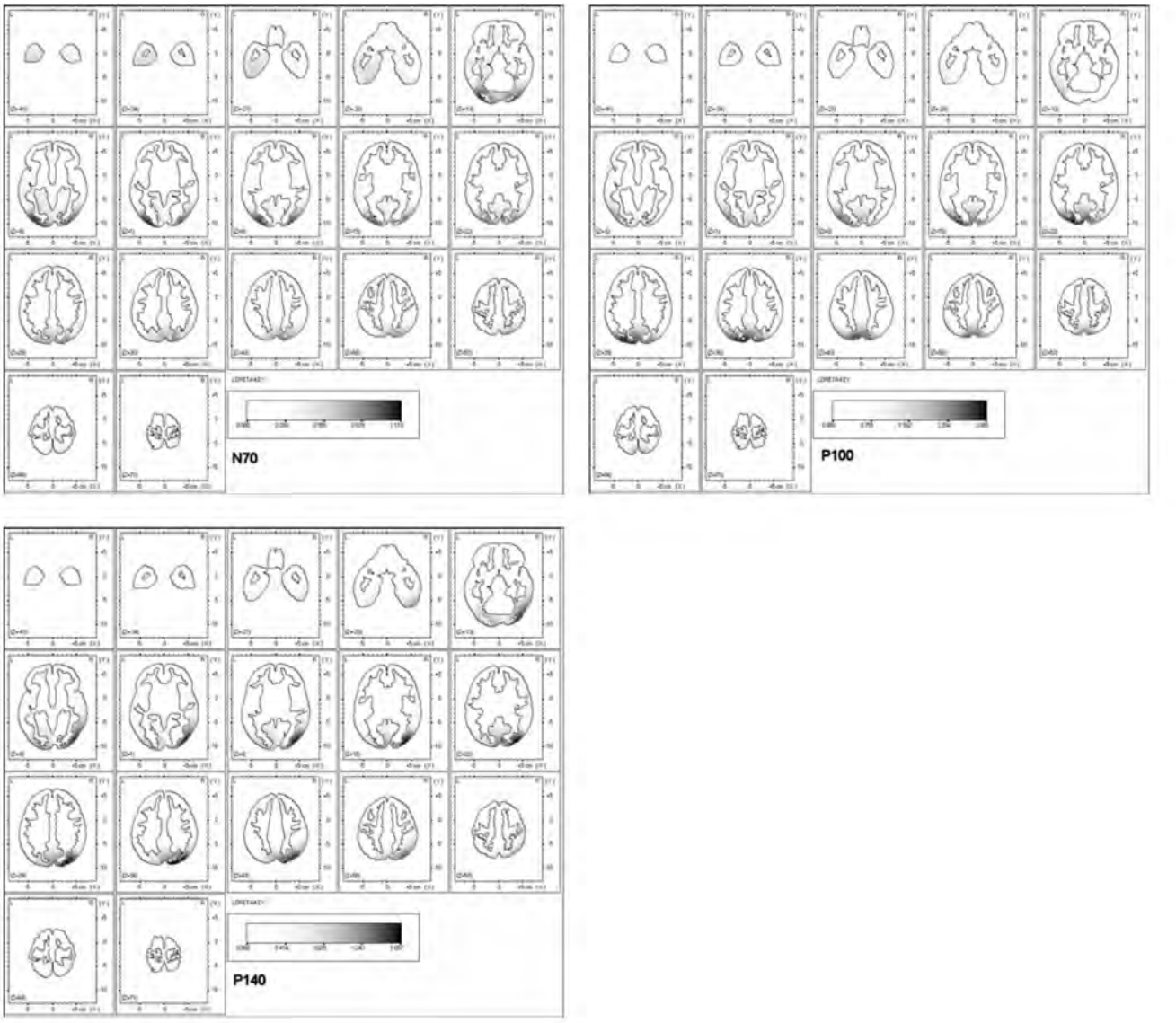

Fig. 7. LORETA-estimated electric neuronal activity (current density, $\mu A/mm^2$) distribution at the main peaks after pattern reversal checkerboard stimulation of the right visual field: N70 (70 msec), P100 (100 msec), and P140 (140 msec). Cortical gray matter is shown outlined in a set of axial (horizontal) slices through the brain, from inferior to superior. Each slice is viewed from above, nose up. Coordinates are: X from left (L) to right (R); Y from posterior to anterior; Z from inferior to superior. The current density is gray-scale encoded.

Fig. 7 illustrates the distribution of electric neuronal activity over cortical gray matter, as estimated with LORETA, at the time instants of the main components (peaks) of the ERP shown in Fig. 6. Note that the maximum activity is localized in left visual cortical areas for the N70 and P100 peaks and then localized in right visual cortical areas for the P140 peak. This is to be expected from known neuroanatomy and neurophysiology of the visual pathway: The right visual field projects onto the right hemiretinas of the eyes and both project into the left visual cortex, which later projects to the right hemisphere using transcallosal connections.

The *ns*NMF model was fitted to the total electric neuronal activity data, which consisted of a positive data matrix of current densities with 2,394 variables (voxels) and 256 objects or items (discrete time instants). In this setting, the basis vectors will be the brain maps of dimension 2,394, corresponding to cortical distributions of electric neuronal activity for different "brain modes," and the encoding vectors will be time series of dimension (discrete duration) 256, corresponding to the time course of activation of the "brain modes." Five ($q = 5$) brain maps and time series were used in this study.

Fig. 8 shows the estimated "encoding vectors," i.e., the time course of activation for each brain mode. In correspondence with known neurophysiology, the three main events in visual information processing are parsimoniously represented separately by sequential activations with peaks at 70 msec for encoding vector "004," followed by the activation peak at 100 msec for encoding vector "005," and ending with the activation peak at 140 msec for encoding vector "002." The time series of encoding vector "001" has maximum amplitude in later stages of visual information processing (> 200 msec). Finally, the time series of encoding vector "003" has the smallest amplitude of all time series and appears to explain only some residual variance at very early and very late stages of visual information processing.

Fig. 9 shows only three "basis vectors" or brain modes, which had current density values large enough to contribute

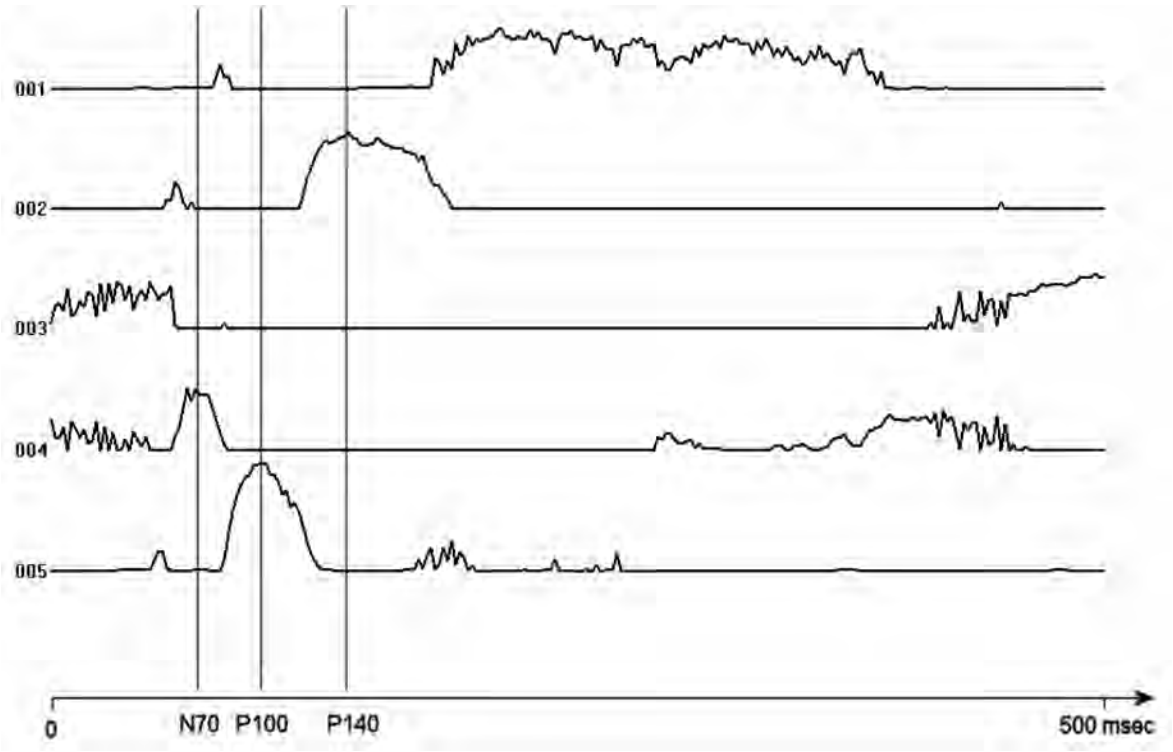

Fig. 8. Encoding vectors, using $\theta = 0.6$ in the *ns*NMF model, corresponding to time course of activation of five "brain modes." Data corresponds to electric neuronal activity for brain response to visual pattern reversal presented to the right visual field. The three vertical cursors correspond to the main components (peaks) of the brain response in this experiment and they are denoted as N70 (70 msec) for encoding vector "004," P100 (100 msec) for encoding vector "005," and P140 (140 msec) for encoding vector "002." Fig. 9 displays only the significant basis vectors corresponding to the brain modes.

sufficiently to the reconstruction of the data. It is worth noting that the time course of activations is extremely sparse, practically segmenting the time axis into disjoint brain modes. When the classical NMF model was applied to this data, activation time series were very smooth and noninterpretable (results not shown here). Furthermore, the only brain modes (basis vectors) that located electric neuronal activity large enough to contribute to the data reconstruction are in satisfactory agreement with known neuroanatomy and neurophysiology of the visual pathway, regarding the timing of activation of the left and right visual cortices.

6 Conclusions and Discussion

There has been great interest in the nonnegative matrix factorization method in the past few years due to its effective ability in extracting human intelligible features. Data analysis processing is a complex task, especially when high dimensional and noisy data is used, so that any method that helps in alleviating the interpretation of the data is more than welcome. The approach presented here is an attempt to improve the ability of the classical NMF algorithm in this process by producing truly sparse components of the data structure and, at the same time, to identify which of these components are better represented by each individual item. Experimental results have shown that the *ns*NMF algorithm described here is capable of achieving this goal. The representation of the basis vectors in the examples presented here shows clear localized features of the data due to the sparseness conditions imposed by the algorithm.

Several real-life applications can benefit from the properties of this method. For example, note that, if the coefficients for each item in the data set are properly sorted, a robust clustering of the items by their most important features can be easily achieved. This process can be interpreted as biclustering [26], [27] since both, the data items (contained in the rows of matrix $\mathbf{H}$) and their features (contained in the columns of matrix $\mathbf{W}$) are grouped together at the same time. The practical advantages of such biclustering methods has been proven in fields such as gene expression data analysis [28], [29], [30], [31], [32], [33], [34], topic extraction from documents [35], [36], and biomedical applications [37], to mention only a few.

The experimental results on both synthetic data and real data sets have shown that the *ns*NMF algorithm outperformed the existing sparse NMF variants in performing parts-based representation of the data while maintaining the goodness of fit. This capability is very useful in real data mining applications where dimensionality reduction can be achieved while the interpretation of the data becomes easier.

Acknowledgments

The authors would like to thank the KEY Foundation for Brain-Mind Research for economic support of this work. This work was also partially supported by the Spanish grants BFU2004-00217/BMC and GR/SAL/0653/2004. A. Pascual-Montano was previously with the KEY Institute for Brain-Mind Research, University Hospital Psychiatry, Zurich and he also acknowledges the support of the Spanish Ramon and Cajal Program.

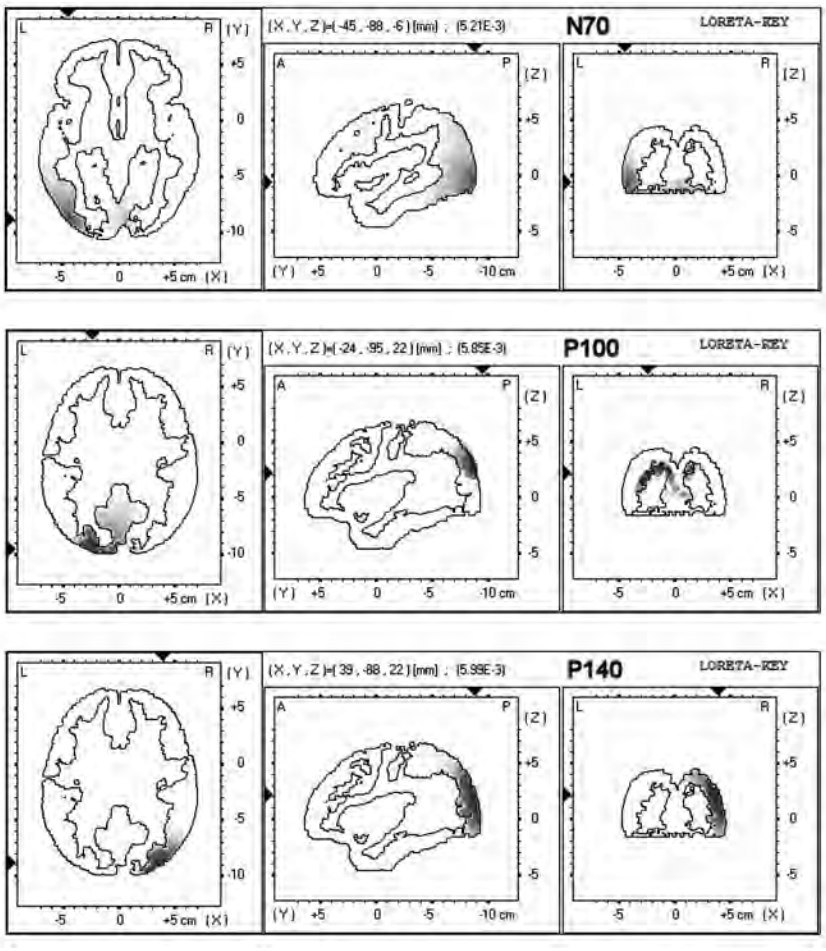

Fig. 9. Basis vectors, using $\theta = 0.6$ in the *ns*NMF model, corresponding to the significant "brain modes." Data corresponds to electric neuronal activity for brain response to visual pattern reversal presented to the right visual field. The three brain maps correspond to the main components (peaks) of the brain response in this experiment and they are denoted as N70 (70 msec) for encoding vector "004," P100 (100 msec) for encoding vector "005," and P140 (140 msec) for encoding vector "002." Fig. 8 displays encoding vectors. The LORETA tomography is displayed in three orthogonal brain views in standard Talairach space, sliced through the region of maximum current density. Left: axial slices, seen from above, nose up; center: sagittal slices, seen from the left; right: coronal slices, seen from the rear. Talairach coordinates: X from left (L) to right (R), Y from posterior (P) to anterior (A), Z from inferior to superior. The locations of the maximum current density are given as (X, Y, Z) coordinates in Talairach space and are graphically indicated by black triangles on the coordinate axes.

REFERENCES

[1] P. Paatero and U. Tapper, "Positive Matrix Factorization—A Nonnegative Factor Model with Optimal Utilization of Error-Estimates of Data Values," *Environmetrics,* vol. 5, pp. 111-126, 1994.

[2] D.D. Lee and H.S. Seung, "Learning the Parts of Objects by Nonnegative Matrix Factorization," *Nature,* vol. 401, pp. 788-791, 1999.

[3] A. Hyvärinen, J. Karhunen, and E. Oja, *Independent Component Analysis.* John Wiley and Sons, 2001.

[4] P.M. Kim and B. Tidor, "Subsystem Identification through Dimensionality Reduction of Large-Scale Gene Expression Data," *Genome Research,* vol. 13, pp. 1706-1718, 2003.

[5] A. Heger and L. Holm, "Sensitive Pattern Discovery with 'Fuzzy' Alignments of Distantly Related Proteins," *Bioinformatics,* vol. 19, no. 1, pp. 130-137, 2003.

[6] P. Paatero, P.K. Hopke, J. Hoppenstock, and S.I. Eberly, "Advanced Factor Analysis of Spatial of PM2.5 in the Eastern United States," *Environ Science Technology,* vol. 37, pp. 2460-2476, 2003.

[7] J.P. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov, "Metagenes and Molecular Pattern Discovery Using Matrix Factorization," *Proc. Nat'l Academy of Science USA,* vol. 101, pp. 4164-4169, 2004.

[8] D. Guillamet and J. Vitria, "Nonnegative Matrix Factorization for Face Recognition," *Proc. Conf. Topics in Artificial Intelligence,* pp. 336-344, 2002.

[9] R. Ramanath, W.E. Snyder, and H. Qi, "Eigenviews for Object Recognition in Multispectral Imaging Systems," *Proc. Applied Imagery Pattern Recognition (AIPR) Workshop,* 2003.

[10] G. Buchsbaum and O. Bloch, "Color Categories Revealed by Nonnegative Matrix Factorization of Munsell Color Spectra," *Vision Research,* vol. 42, pp. 559-563, 2002.

[11] R. Ramanath, R.G. Kuehni, W.E. Snyder, and D. Hinks, "Spectral Spaces and Color Spaces," *Color Research and Application,* vol. 29, pp. 29-37, 2004.

[12] P. Smaragdis and J.C. Brown, "Nonnegative Matrix Factorization for Polyphonic Music Transcription," *Proc. IEEE Workshop Applications of Signal Processing to Audio and Acoustics,* 2003.

[13] B.W. Mel, "Computational Neuroscience. Think Positive to Find Parts," *Nature,* vol. 401, pp. 759-760, 1999.

[14] T. Feng, S.Z. Li, H.-Y. Shum, and H. Zhang, "Local Nonnegative Matrix Factorization as a Visual Representation," *Proc. Second Int'l Conf. Development and Learning (ICDL '02),* 2002.

[15] P.O. Hoyer, "Nonnegative Sparse Coding," *Proc. IEEE Workshop Neural Networks for Signal Processing,* 2002.

[16] P.O. Hoyer, "Nonnegative Matrix Factorization with Sparseness Constraints," *J. Machine Learning Research,* vol. 5, pp. 1457-1469, 2004.

[17] W. Liu, N. Zheng, and X. Lu, "Nonnegative Matrix Factorization for Visual Coding," *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing (ICASSP 2003),* 2003.

[18] B.A. Olshausen and D.J. Field, "Emergence of Simple-Cell Receptive Field Properties by Learning a Sparse Code for Natural Images," *Nature,* vol. 381, pp. 607-609, 1996.

[19] G.F. Harpur and R.W. Prager, "Development of Low Entropy Coding in a Recurrent Network," *Network: Computation in Neural Systems,* vol. 7, pp. 277-284, 1996.
[20] D. Donoho and V. Stodden, "When Does Nonnegative Matrix Factorization Give a Correct Decomposition into Parts?" *Proc. 17th Ann. Conf. Neural Information Processing Systems (NIPS 2003),* 2003.
[21] MIT Center For Biological and Computation Learning, CBCL Face Database #1, http://www.ai.mit.edu/projects/cbcl, 2005.
[22] R.D. Pascual-Marqui, C.M. Michel, and D. Lehmann, "Low Resolution Electromagnetic Tomography: A New Method for Localizing Electrical Activity in the Brain," *Int'l J. Psychophysiology,* vol. 18, pp. 49-65, 1994.
[23] J. Yao and J.P. Dewald, "Evaluation of Different Cortical Source Localization Methods Using Simulated and Experimental EEG Data," *Neuroimage,* vol. 25, pp. 369-382, 2005.
[24] C. Mulert, L. Jager, R. Schmitt, P. Bussfeld, O. Pogarell, H.J. Moller, G. Juckel, and U. Hegerl, "Integration of fMRI and Simultaneous EEG: Towards a Comprehensive Understanding of Localization and Time-Course of Brain Activity in Target Detection," *Neuroimage,* vol. 22, pp. 83-94, 2004.
[25] R.D. Pascual-Marqui, "Review of Methods for Solving the EEG Inverse Problem," *Int'l J. Bioelectromagnetism,* vol. 1, pp. 75-86, 1999.
[26] S.C. Madeira and A.L. Oliveira, "Biclustering Algorithms for Biological Data Analysis: A Survey," *IEEE/ACM Trans. Computational Biology and Bioinformatics,* vol. 1, pp. 24-45, 2004.
[27] A. Tanay, R. Sharan, and R. Shamir, "Biclustering Algorithms: A Survey," *Handbook of Computational Molecular Biology,* in press.
[28] Q. Sheng, Y. Moreau, and B. DeMoor, "Biclustering Microarray Data by Gibbs Sampling," *Bioinformatics,* vol. 19, no. 2, pp. 196-205, 2003.
[29] Y. Kluger, R. Basri, J.T. Chang, and M. Gerstein, "Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions," *Genome Research,* vol. 13, pp. 703-716, 2003.
[30] A. Tanay, R. Sharan, and R. Shamir, "Discovering Statistically Significant Biclusters in Gene Expression Data," *Bioinformatics,* vol. 18, no. 1, pp. 136-144, 2002.
[31] Y. Cheng and G.M. Church, "Biclustering of Expression Data," *Proc. Int'l Conf. Intellegent Systems Molecular Biology,* vol. 8, pp. 93-103, 2000.
[32] H. Turner, T. Bailey, and W. Krzanowski, "Improved Biclustering of Microarray Data Demonstrated through Systematic Performance Tests," *Computational Statistics and Data Analysis,* vol. 48, pp. 235-254, 2005.
[33] Y. Kluger, R. Basri, J.T. Chang, and M. Gerstein, "Spectral Biclustering of Microarray Data: Coclustering Genes and Conditions," *Genome Research,* vol. 13, pp. 703-716, 2003.
[34] M. Dugas, S. Merk, S. Breit, and P. Dirschedl, "Mdclust—Exploratory Microarray Analysis by Multidimensional Clustering," *Bioinformatics,* vol. 20, pp. 931-936, 2004.
[35] S.H. Srinivasan, "Features for Unsupervised Document Classification," *Proc. Sixth Workshop Computational Language Learning (CoNLL-2002),* 2002.
[36] J.-H. Chang, J.W. Lee, Y. Kim, and B.-T. Zhang, "Topic Extraction from Text Documents Using Multiple-Cause Networks," *Proc. Seventh Pacific Rim Int'l Conf. Artificial Intelligence: Trends in Artificial Intelligence,* 2002.
[37] I. VanMechelen, H.H. Bock, and P. DeBoeck, "Two-Mode Clustering Methods: A Structured Overview," *Statistical Methods in Medical Research,* vol. 13, pp. 363-394, 2004.

Alberto Pascual-Montano received the doctoral degree in computer science from the Universidad Autonoma in Madrid, Spain. He is an assistant professor in the Computer Architecture Department of the Complutense University in Madrid. He worked at the National Center of Biotechnology from 1998 to 2004 researching in topics like bioinformatics, pattern recognition, and image processing. In 2004, he joined the KEY Institute for Brain-Mind Research in Zurich, for a postdoctoral stay. His research interests include pattern recognition, image processing, bioinformatics, and computer architecture. He is a member of the IEEE.

J.M. Carazo received the MS degree in theoretical physics and the PhD degree in molecular biology. He is a senior research scientist of the Spanish Research Council, CSIC, where he directs the Biocomputing Unit of the National Center for Biotechnology in Madrid. He worked at the IBM Madrid Scientific Center from 1981 to 1986 and from 1987 to 1989 at the Howard Hughes Medical Center at the New York State Health Department in Albany before joining the CSIC and the CNB in 1989. His research interests are in the area of multidimensional image classification and three-dimensional volume reconstruction from electron microscopy projection images. He has published more than 120 papers in biological and engineering journals and directed large international projects in the area of biological multidimensional databases. He is a senior member of the IEEE.

Kieko Kochi received doctoral degrees in medicine, neurology, and psychiatry. She is the president of the KEY Foundation for Brain-Mind Research, and director of the KEY Institute for Brain-Mind Research.

Dietrich Lehmann received the doctoral degree in medicine from the University of Heidelberg and an honorary doctoral degree from the University of Jena. He is a professor emeritus of clinical neurophysiology at the University of Zürich. After clinical neurology and research in neurophysiology at the Universities of Heidelberg, Munich, and Freiburg i.Br., in 1961, he joined the University of California at Los Angeles and, later, the Department of Visual Sciences, University of the Pacific, San Francisco. He moved to Zürich in 1971, where in 1995, after his retirement, he became the first director of the newly founded KEY Institute for Brain-Mind Research at the University Hospital of Psychiatry in Zürich. Since 1998, he has been a senior research scientist at this institute. His work centers around the development of multichannel evoked (ERP) and spontaneous (EEG) brain electric field mapping and spatial analysis, including source analysis and temporal microstate analysis and applying these tools to the study of human brain electric mechanisms of normal and pathological perception, cognition, and emotion.

Roberto D. Pascual-Marqui received the PhD degree in biological sciences from the Cuban Neuroscience Center in 1988. He was the head of the Neurophysics Laboratory at the Cuban Neuroscience Center from 1981-1992. In 1992, he joined the Brain Mapping Laboratory, Department of Neurology, University Hospital, Zürich. Since 1996, he has been a senior research scientist at the KEY Institute for Brain-Mind Research, University Hospital of Psychiatry, Zürich. He received the "Privatedozent" degree in 2004 from the Faculty of Medicine, University of Zürich. His research interests include the development of techniques for functional mapping of the human brain based on EEG and MEG, the analysis of the spatio-temporal properties of brain electric activity and its relation to the mind (cognition), and the development of tools for pattern recognition (in general).