Andrzej Cichocki, Rafal Zdunek,
and Shun-ichi Amari

Nonnegative Matrix and Tensor Factorization

There has been a recent surge of interest in matrix and tensor factorization (decomposition), which provides meaningful latent (hidden) components or features with physical or physiological meaning and interpretation. Nonnegative matrix factorization (NMF) and its extension to three-dimensional (3-D) nonnegative tensor factorization (NTF) attempt to recover hidden nonnegative common structures or patterns from usually redundant data. These lecture notes present several alternative methods for solving NMF/NTF problems.

RELEVANCE

The topics presented here extend the standard NMF algorithms [1] to the alternative more robust algorithms that may find more practical applications within data analysis (pattern recognition, segmentation, clustering, dimensionality reduction, text mining, language modeling) [1], [2]; signal and image processing (blind source separation, spectra recovering, music transcription) [3], [4]; and neurobiology (gene separation, EEG data analysis) [5].

PREREQUISITES

The prerequisites for understanding and using these lecture notes consist of basic optimization theory (gradient descent and quasi-Newton methods, Karush-Kuhn-Tucker (KKT) conditions) and basic linear algebra. Information theory could be also useful but not necessary.

PROBLEM STATEMENT

The underlaying model used in NMF has the following nonnegatively constrained linear form:

$$Y = AX + N, \tag{1}$$

where $Y = [y_{ik}] \in \mathbb{R}^{I \times K}$ is a matrix of observations, $A = [a_{ij}] \in \mathbb{R}_+^{I \times J}$ is an unknown nonnegative basis matrix, $X = [x_{jk}] \in \mathbb{R}_+^{J \times K}$ is a matrix of unknown hidden nonnegative components, and $N = [n_{ik}] \in \mathbb{R}^{I \times K}$ is a matrix of noise or errors. Here and elsewhere, $\mathbb{R}_+$ denotes the nonnegative orthant (subspace) with appropriate dimensions. The low rank denoted as J is assumed to be known, or it can be easily estimated with singular value decomposition (SVD). In many applications: $K \gg I \geq J$.

Basically, the factorization is achieved by alternating minimization of a suitable cost function, or optionally a set of cost functions, subject to nonnegativity constraints. Due to intrinsic indeterminacies of the alternating minimization, other possible natural constraints such as sparsity, smoothness, boundness, or unimodality are often assumed to recover the true components. The cost function to be minimized can be determined on the basis of a prior knowledge on statistical distribution of noise.

COST FUNCTIONS: GENERALIZED DIVERGENCES

For normally distributed noisy disturbances in (1), the regularized squared Euclidean distance

$$D_F(Y||AX) = \frac{1}{2}\|Y - AX\|_F^2 + \alpha_A J_A(A) + \alpha_X J_X(X) \quad \text{with} \quad a_{ij} \geq 0, \quad x_{jk} \geq 0, \quad \forall\, i, j, k, \tag{2}$$

is an optimal cost function, where $\|\cdot\|_F$ denotes the Frobenius norm, α_A and α_X are nonnegative regularization parameters, and the terms $J_X(X)$, $J_A(A)$ are used to enforce certain application-dependent characteristics of a desired solution. As special practical case, we have $J_X(X) = \sum_{jk} f(x_{jk})$ where $f(\cdot)$ is a suitably chosen function that measures smoothness or sparsity. To achieve sparse representations we usually choose $f(x_{jk}) = x_{jk}$ for $x_{jk} \geq 0$, or the diversity measure $J_X(X) = \sum_{k=1}^{K} (\sum_{j=1}^{J} x_{jk})^2 = \text{trace}\,\{X^T E X\}$, where $E \in \mathbb{R}^{J \times J}$ is a matrix of all ones [6].

For non-Gaussian distributed noise we may use the Cisiszár, Bregman, α- or β-divergence [4], [7]. For example, the α-divergence, which generalizes the Kullback-Leibler I-divergence can be expressed as

$$D_A^{(\alpha)}(Y||AX) = \frac{1}{\alpha(\alpha-1)} \sum_{ik} \Big(y_{ik}^{\alpha} [AX]_{ik}^{1-\alpha} - \alpha\, y_{ik} + (\alpha - 1)\, [AX]_{ik} \Big), \quad \alpha \neq 0, \tag{3}$$

where $[AX]_{ik}$ stands for the ikth entry of AX.

We recall here that as special cases of the α-divergence for $\alpha = 2, 0.5, -1$, we obtain the Pearson's chi-square and Hellinger's and Neyman's chi-square distances, respectively, while for the cases $\alpha = 1$ and $\alpha = 0$ the divergence has to be defined by the limit points. When these limits are evaluated one obtains the generalized Kullback-Leibler divergence (I-divergence) for $\alpha \to 1$, and the dual Kullback-Leibler divergence for $\alpha \to 0$.

Digital Object Identifier 10.1109/MSP.2007.911394

1053-5888/08/$25.00©2008IEEE

To perform the alternating minimization of a given cost function, we can select one of the following methods: multiplicative, projected gradient, fixed point alternating least squares (ALS), and quasi-Newton.

METHOD 1: MULTIPLICATIVE ALGORITHMS

The multiplicative update rules are the most commonly used for NMF [1], [2]. For example, by applying the standard gradient descent technique to the cost function (2) and selecting suitable learning rates we obtain the algorithm that is an extended version of the image space reconstruction algorithm (ISRA) [1], [4]:

$$a_{ij} \leftarrow a_{ij} \frac{\left[[YX^T]_{ij} - \alpha_A \Psi_A(a_{ij})\right]_+}{[AXX^T]_{ij} + \varepsilon},$$

$$x_{jk} \leftarrow x_{jk} \frac{\left[[A^T Y]_{jk} - \alpha_X \ \Psi_X(x_{jk})\right]_+}{[A^T AX]_{jk} + \varepsilon}, \tag{4}$$

where the nonlinear operator is defined as $[x]_+ = \max\{\varepsilon, x\}$ with small ε to avoid numerical instabilities, and $\Psi_A(a_{ij}) = \partial J_A(\boldsymbol{A})/\partial a_{ij}$, $\Psi_X(x_{jk}) = \partial J_X(\boldsymbol{X})/\partial x_{jk}$. Similarly applying the gradient descent technique to the α-divergence we can derive alternative NMF multiplicative rules [4], [8]:

$$a_{ij} \leftarrow a_{ij} \left(\frac{\sum_{k=1}^{K} (y_{ik}/[AX]_{ik})^{\alpha} x_{jk}}{\sum_{k=1}^{K} x_{jk}} \right)^{1/\alpha},$$

$$x_{jk} \leftarrow x_{jk} \left(\frac{\sum_{i=1}^{I} a_{ij} (y_{ik}/[AX]_{ik})^{\alpha}}{\sum_{i=1}^{I} a_{ij}} \right)^{1/\alpha}, \tag{5}$$

with normalization of the columns of $\boldsymbol{A}$ in each alternating step to unit length: $a_{ij} \leftarrow a_{ij}/\sum_p a_{pj}$. Multiplicative algorithms are relatively simple and usually parameter free, however, their convergence speed is relatively slow, especially for large-scale problems. Thus they can be mostly useful only for small- and medium-scale problems, and if representations of signals or data are very sparse.

METHOD 2: PROJECTED GRADIENT

The projected gradient method [9] can be generally expressed by iterative updates:

$$\begin{aligned} \boldsymbol{A} &\leftarrow [\boldsymbol{A} - \eta_A \boldsymbol{P}_A]_+, \\ \boldsymbol{X} &\leftarrow [\boldsymbol{X} - \eta_X \boldsymbol{P}_X]_+, \end{aligned} \tag{6}$$

where $\boldsymbol{P}_A$ and $\boldsymbol{P}_X$ are descent directions for $\boldsymbol{A}$ and $\boldsymbol{X}$, respectively. There are several rules for choosing the adaptive learning rates η_A and η_X. For example, in the interior point gradient (IPG) algorithm [8]:

$$\begin{aligned} \boldsymbol{A} &\leftarrow \boldsymbol{A} - \eta_A \, [\boldsymbol{A} \ \oslash \ (\boldsymbol{A}\boldsymbol{X}\boldsymbol{X}^T)] \\ &\quad \odot \ [(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})\boldsymbol{X}^T], \end{aligned} \tag{7}$$

$$\begin{aligned} \boldsymbol{X} &\leftarrow \boldsymbol{X} - \eta_X \, [\boldsymbol{X} \oslash (\boldsymbol{A}^T \boldsymbol{A}\boldsymbol{X})] \\ &\quad \odot \ [\boldsymbol{A}^T(\boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X})], \end{aligned} \tag{8}$$

where $\odot$ and $\oslash$ mean component-wise multiplication and division, respectively, the positive learning rates $\eta_A > 0$ and $\eta_X > 0$ are adjusted in each iterative step—on the one hand, to be close to η_A^* and η_X^*, which are the exact minimizers of $D_F(\boldsymbol{Y}||(\boldsymbol{A} - \eta_A \boldsymbol{P}_A)\boldsymbol{X})$ and $D_F(\boldsymbol{Y}||\boldsymbol{A}(\boldsymbol{X} - \eta_X \boldsymbol{P}_X))$, respectively, and on the other hand, to keep some distance to a boundary of the nonnegative orthant.

Usually, the projected gradient algorithms are more efficient and provide much better performance than the multiplicative algorithms, especially, when using the multilayer NMF approach [4], [8], [10].

METHOD 3: FIXED POINT ALS ALGORITHMS

Fixed point alternating least squares (FP-ALS) algorithms [10] do not use directly the gradient descent approach but rather attempt to establish iterative algorithms based on the KKT conditions. For example, by computing the gradients of the cost function (2), equating them to zero, and assuming that we enforce the nonnegativity constraints with a simple "half-rectifying" nonlinear projection, we have the following FP-ALS algorithm:

$$\begin{aligned} \boldsymbol{A} &\leftarrow [(\boldsymbol{Y}\boldsymbol{X}^T - \alpha_A \Psi_A(\boldsymbol{A}))(\boldsymbol{X}\boldsymbol{X}^T)^{-1}]_+, \\ \boldsymbol{X} &\leftarrow [(\boldsymbol{A}^T\boldsymbol{A})^{-1}(\boldsymbol{A}^T\boldsymbol{Y} - \alpha_X \Psi_X(\boldsymbol{X}))]_+, \end{aligned} \tag{9}$$

where $\Psi_A(\boldsymbol{A}) = \partial J_A(\boldsymbol{A})/\partial \boldsymbol{A}$, $\Psi_X(\boldsymbol{X}) = \partial J_X(\boldsymbol{X})/\partial \boldsymbol{X}$.

Instead of minimizing the global cost function we can minimize the set of local functions defined as

$$\begin{aligned} D_F^{(j)}(\boldsymbol{Y}^{(j)} \| \boldsymbol{a}_j \underline{\boldsymbol{x}}_j) &= \frac{1}{2} \| (\boldsymbol{Y}^{(j)} - \boldsymbol{a}_j \underline{\boldsymbol{x}}_j) \|_F^2 \\ &\quad + \alpha_A^{(j)} J_A(\boldsymbol{a}_j) \\ &\quad + \alpha_X^{(j)} J_X(\underline{\boldsymbol{x}}_j), \\ &\quad (j = 1, 2, \ldots, J), \end{aligned} \tag{10}$$

where

$$\begin{aligned} \boldsymbol{Y}^{(j)} &= \boldsymbol{Y} - \sum_{r \neq j} \boldsymbol{a}_r \underline{\boldsymbol{x}}_r \\ &= \boldsymbol{Y} - \boldsymbol{A}\boldsymbol{X} + \boldsymbol{a}_j \underline{\boldsymbol{x}}_j, \end{aligned} \tag{11}$$

$\boldsymbol{a}_j \in \mathbb{R}_+^{I \times 1}$ are columns of $\boldsymbol{A}$, $\underline{\boldsymbol{x}}_j \in \mathbb{R}_+^{1 \times K}$ are rows of $\boldsymbol{X}$. The construction of such a set of local cost functions follows from the simple observation that the observed data can be decomposed approximately as $\boldsymbol{Y} = \sum_{j=1}^{J} \boldsymbol{a}_j \underline{\boldsymbol{x}}_j$. From stationary points and under sparsity assumption, i.e., $J_A(\boldsymbol{a}_j) = \|\boldsymbol{a}_j\|_1$ and $J_X(\underline{\boldsymbol{x}}_j) = \|\underline{\boldsymbol{x}}_j\|_1$, we obtain a new set of sequential learning rules, which we call the hierarchical ALS (HALS) algorithm [10]:

$$\begin{aligned} \boldsymbol{a}_j &\leftarrow \frac{1}{\underline{\boldsymbol{x}}_j \underline{\boldsymbol{x}}_j^T} \left[\boldsymbol{Y}^{(j)} \underline{\boldsymbol{x}}_j^T - \alpha_A^{(j)} \right]_+, \\ \underline{\boldsymbol{x}}_j &\leftarrow \frac{1}{\boldsymbol{a}_j^T \boldsymbol{a}_j} \left[\boldsymbol{a}_j^T \boldsymbol{Y}^{(j)} - \alpha_X^{(j)} \right]_+, \\ &(j = 1, 2, \ldots, J). \end{aligned} \tag{12}$$

It is interesting to note that such nonlinear projections can be imposed individually for each component (source) $\underline{\boldsymbol{x}}_j$ and/or vector $\boldsymbol{a}_j$, so the algorithm can be directly extended to a semi-NMF or a semi-NTF model in which some parameters are assumed to be bipolar. Moreover, in practice, it is necessary to normalize in each iterative step the column vectors $\boldsymbol{a}_j$ or the row vectors $\underline{\boldsymbol{x}}_j$ to unit length vectors [in the sense of l_p norm $(p = 1, 2, \ldots, \infty)$]. In the special case of the l_2 norm, the above algorithm can be further simplified by neglecting the corresponding denominators in (12).

This is a robust extension of the standard FP-ALS algorithm, which allows us to estimate sparse nonnegative components even when the number of observations is less than the number of sources.

The FP-ALS algorithms are very fast and suitable for large-scale problems. Moreover, they can be easily adapted to semi-NMF (when only some set of parameters are nonnegative), tri-NMF models ($Y = AWX$), and generalized NMF with additional constraints such as orthogonality or sparsity.

METHOD 4: QUASI-NEWTON ALGORITHM

If A is estimated with solving a highly over-determined system of linear equations: $X^T A^T = Y^T$, i.e., for $K \gg J$, a very efficient approach is to use the quasi Newton (QN) method [6], [10] that exploits the information on the curvature of the cost function. The Hessian $\nabla_A^2(D_F) = I_I \otimes XX^T \in \mathbb{R}^{IJ \times IJ}$ of $D_F(Y||AX)$ has a block diagonal structure with the same blocks, and hence, we can simplify the update rule for A with the Newton method to the following form:

$$A \leftarrow \left[A - \nabla_A(D_F(Y\|AX))H_A^{-1}\right]_+, \tag{13}$$

where $\nabla_A D_F(Y\|AX) = (AX - Y)X^T \in \mathbb{R}^{I \times J}$, and $H_A = XX^T \in \mathbb{R}^{J \times J}$. The matrix H_A may be ill-conditioned, especially if X is sparse, and due to this the Levenberg-Marquardt approach is used to control the ill-conditioning of the Hessian. Thus we have the following update for A:

$$A \leftarrow \left[A - (AX - Y)X^T(XX^T + \lambda I_J)^{-1}\right]_+, \tag{14}$$

where λ is a damping parameter, and $I_J \in \mathbb{R}^{J \times J}$ is an identity matrix.

Since the alternating minimization rule in NMF is not convex, the selection of initial conditions is very important. To minimize the risk of getting trapped in local minima of the cost functions, we use some steering technique that comes from a simulated annealing approach. The solution is triggered with the exponential rule: $\lambda \leftarrow \lambda_0 \exp\{-\tau s\}$, where s is an index of a current alternating step, which gradually steers the updates from the steepest descent to Newton-like.

METHOD 5: MULTILAYER TECHNIQUE

To improve the performance of the NMF algorithms, especially for ill-conditioned and badly scaled data, and also to reduce the risk of getting stuck in local minima of a cost function subject to nonconvex alternating minimization rule, we have developed a simple hierarchical multistage procedure [4], [8], [10] combined with multistart initialization, in which we perform a sequential decomposition of nonnegative matrices as follows.

In the first step, we perform the basic decomposition $Y \approx A^{(1)}X^{(1)} \in \mathbb{R}^{I \times K}$ using any available NMF algorithm. In the second stage, the results obtained from the first stage are used to build up a new matrix $Y \leftarrow X^{(1)}$, that is, in the next step, we perform the similar decomposition $X^{(1)} \approx A^{(2)}X^{(2)} \in \mathbb{R}^{J \times K}$, using the same or different update rules. We continue our decomposition taking into account only the last achieved components. The process can be repeated arbitrarily many times until some stopping criteria are satisfied.

Thus, our multilayer NMF model has the form:

$$Y \approx A^{(1)}A^{(2)} \cdots A^{(L)}X^{(L)}, \tag{15}$$

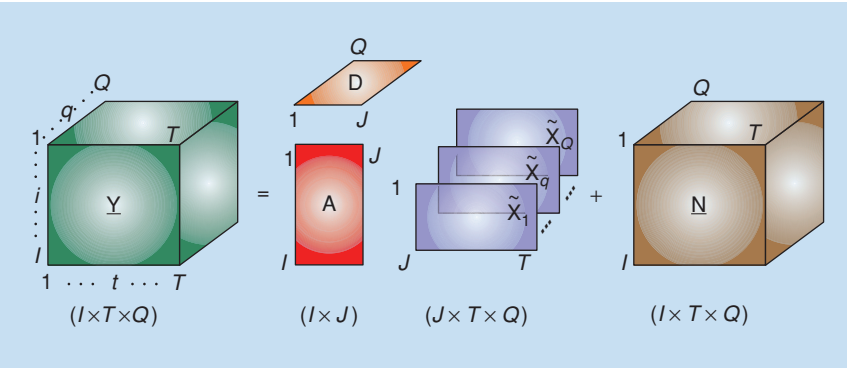
with final results $A = A^{(1)}A^{(2)} \cdots A^{(L)}$ and $X = X^{(L)}$. Physically, this means that we build up a distributed system that has many layers or cascade connections of L mixing subsystems. The key point in our approach is that the learning (update) process to find the parameters of matrices $X^{(l)}$ and $A^{(l)}$ is performed sequentially, i.e., layer by layer, where each layer is randomly initialized with different initial conditions. In fact, we found that the hierarchical multilayer approach is generally necessary to apply to achieve high performance for all the proposed algorithms.

EXTENSION TO NTF

The two-dimensional (2-D) NMF models discussed earlier can be extended to 3-D NTF models. Alternatively, some 3-D NTF models can be reduced to the 2-D NMF model. For simplicity we discuss briefly this latter case. We consider an example of 3-D NTF model (referred here to as the NTF2 model) [8] that is illustrated in Figure 1 and described as follows,

$$Y_q = AD_q\widetilde{X}_q + N_q = AX_q + N_q, \quad (q = 1, 2, \ldots, Q), \tag{16}$$

where $Y_q = [y_{itq}] \in \mathbb{R}^{I \times T}$ is a qth frontal slice (matrix) of the observed (known) 3-D data (signal) tensor $\underline{Y} \in \mathbb{R}^{I \times T \times Q}$, $D_q \in \mathbb{R}_+^{J \times J}$ is a diagonal scaling matrix that holds the qth row of the matrix $D \in \mathbb{R}^{Q \times J}$, $A = [a_{ij}] = [a_1, a_2, \ldots, a_J] \in \mathbb{R}^{I \times J}$ is a mixing or basis matrix, $\widetilde{X}_q = [\tilde{x}_{jtq}] \in \mathbb{R}^{J \times T}$ repre-



[FIG1] **Illustration of the NTF2 model: decomposition of 3D tensor into a set of nonnegative matrices: $A, D, \{\widetilde{X}_q\}$. In the special case, the model is reduced to the standard NTF model for $\widetilde{X}_q = \widetilde{X}_1, \forall q$, and to tri-NMF model for $Q = 1$.**

[TABLE 1] SIR [dB] AND RET FOR THE NMF ALGORITHMS DISCUSSED EARLIER.

LAYERS	$L = 1$				$L = 3$			
ALGORITHM	**WORST SIR**	**MEAN SIR**	**BEST SIR**	**RET**	**WORST SIR**	**MEAN SIR**	**BEST SIR**	**RET**
FP-ALS FOR $\boldsymbol{A}$ AND $\boldsymbol{X}$	15.1	35	60.2	1	32	70.1	135.2	2.91
FP-ALS FOR $\boldsymbol{A}$, HALS FOR $\boldsymbol{X}$	13	29	51.4	1.85	35	57.2	108.8	5.08
QN FOR $\boldsymbol{A}$ AND FP-ALS FOR $\boldsymbol{X}$	81	90.3	92.8	1.13	89.7	96.2	99.4	3.21
QN FOR $\boldsymbol{A}$ AND HALS FOR $\boldsymbol{X}$	35.2	35.2	35.2	2.43	31.5	31.5	31.5	6.13
ISRA FOR $\boldsymbol{A}$ AND $\boldsymbol{X}$	5.8	16.7	26.6	1.05	5	28.5	44.7	2.96
IPG FOR $\boldsymbol{A}$ AND $\boldsymbol{X}$	14.9	22.2	31.3	2.67	24.2	44.1	60.2	4.85

sents unknown normalized sources or hidden components in qth slice, $X_q = D_q \widetilde{X}_q = [x_{jtq}] \in \mathbb{R}^{J \times T}$ represents re-normalized (differently scaled) sources, and $N_q = [n_{itq}] \in \mathbb{R}^{I \times T}$ represents the q-th frontal slice of the tensor $\underline{N} \in \mathbb{R}^{I \times T \times Q}$ representing noise or errors, depending on applications.

Using row-wise unfolding the model (16) can be represented equivalently by one single matrix equation (1), where $Y = [y_{ik}] = [Y_1, Y_2, \ldots, Y_Q] \in \mathbb{R}^{I \times K}$, $X = [x_{jk}] = [X_1, X_2, \ldots, X_Q] \in \mathbb{R}^{J \times K}$, and $N = [n_{ik}] = [N_1, N_2, \ldots, N_Q] \in \mathbb{R}^{I \times K}$ are block row-wise unfolded matrices with $K = QT$.

NUMERICAL RESULTS

All the NMF algorithms presented here have been extensively tested using many difficult benchmarks for signals and images with various statistical distributions and additive noise and also with real EEG data. The simulation results given next have been obtained for the benchmark used in [6], in which five nonnegative signals have been mixed with dense randomly generated matrix $A \in \mathbb{R}^{10 \times 5}$ with an uniform distribution.

The results have been evaluated using the standard signal-to-interference ratio (SIR) [decibels] and the relative elapsed time (RET), which was obtained by normalizing the ET of each algorithm to the ET of the FP-ALS algorithm. The implementation and speed evaluation have been performed using the MATLAB toolbox in [4], [8], and [10]. On an Intel Pentium 4, CPU 3.6 GHz, 4 GB RAM computer, the FP-ALS algorithm (used as a reference) required 2 s.

The mean SIRs for estimation of the sources have been calculated from 100 Monte Carlo (MC) samples. In each MC run, the learning process is initialized randomly and terminated after 1,000 alternating steps. For the QN method, we set heuristically $\lambda_0 = 100$ and $\tau = 0.02$. The number of layers in the multilayer technique is denoted by L.

The difference in the performance stems from convergence properties of the algorithms (the second-order methods converge much faster than the multiplicative methods), additional regularization terms (sparsification or smoothing strongly restrict the area of feasible solutions), the exponential rule in the QN method (which steers the updates to the global solution), and the multi-initialization with the multilayer technique (which selects better initial conditions).

CONCLUSIONS: WHAT WE HAVE LEARNED

In these lecture notes we have outlined several approaches to solve a NMF/NTF problem. The following main conclusions can be drawn:

1) Multiplicative algorithms are not necessary the best approaches for NMF, especially if data representations are not very redundant or sparse.
2) Much better performance can be achieved using the FP-ALS (especially for large-scale problems), IPG, and QN methods.
3) To achieve high performance it is quite important to use the multilayer structure with multistart initialization conditions.
4) To estimate physically meaningful nonnegative components it is often necessary to use some a priori knowledge and impose additional constraints or regularization terms (to control sparsity, boundness, continuity or smoothness of the estimated nonnegative components).

AUTHORS

Andrzej Cichocki (cia@brain.riken.jp) is the head of the Laboratory for Advanced Brain Signal Processing in RIKEN Brain Science Institute (BSI), Japan. He is also a professor with Warsaw University of Technology, and IBS, Polish Academy of Science (PAN), Warsaw, Poland.

Rafal Zdunek (rafal.zdunek@pwr.wroc.pl) is a research scientist in RIKEN BSI, Japan, and a lecturer at the Institute of Telecommunications, Teleinformatics, and Acoustics, Wroclaw University of Technology, Wroclaw, Poland.

Shun-ichi Amari (amari@brain.riken.jp) is a director of RIKEN BSI, Japan.

REFERENCES

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

[2] M.W. Berry, M. Browne, A.N. Langville, V.P. Pauca, and R.J. Plemmons, "Algorithms and applications for approximate nonnegative matrix factorization," *Computational Statistics and Data Analysis*, 2007, in press: doi:10.1016/j.csda.2006.11.006.

[3] P. Sajda, S. Du, T.R. Brown, R. Stoyanova, D.C. Shungu, X. Mao, and L.C. Parra, "Nonnegative matrix factorization for rapid recovery of constituent spectra in magnetic resonance chemical shift imaging of the brain," *IEEE Trans. Medical Imaging*, vol. 23, no. 12, pp. 1453–1465, 2004.

[4] A. Cichocki, S. Amari, R. Zdunek, R. Kompass, G. Hori, and Z. He, "Extended SMART algorithms for non-negative matrix factorization," *Springer LNAI*, vol. 4029, pp. 548–562, 2006.

[5] J.P. Brunet, P. Tamayo, T.R. Golub, and J.P. Mesirov, "Metagenes and molecular pattern discovery using matrix factorization," *PNAS*, vol. 101, no. 12, pp. 4164–4169, 2000.

[6] R. Zdunek and A. Cichocki, "Nonnegative matrix factorization with constrained second-order optimization," *Signal Processing*, vol. 87, no. 8, pp. 1904–1916, 2007.

[7] I. Dhillon and S. Sra, "Generalized nonnegative matrix approximations with Bregman divergences," in *Proc. Advances in Neural Information Processing Systems (NIPS)*, Vancouver, Canada, Dec. 2005, pp. 283–290.

[8] A. Cichocki, R. Zdunek, S. Choi, R. Plemmons, and S.-I. Amari, "Novel multi-layer nonnegative tensor factorization with sparsity constraints," *Springer LNCS*, vol. 4432, pp. 271–280, 2007.

[9] C.-J. Lin, "Projected gradient methods for nonnegative matrix factorization," *Neural Computation*, vol. 19, no. 10, pp. 2756-2779, 2007.

[10] A. Cichocki, R. Zdunek, and S.-I. Amari, "Hierarchical ALS algorithms for nonnegative matrix and 3D tensor factorization," *Springer LNCS*, vol. 4666, pp. 169–176, 2007.

SP