Sparse Super Symmetric Tensor Factorization

Andrzej CICHOCKI*, Marko JANKOVIC, Rafal ZDUNEK**
, and Shun-ichi AMARI

RIKEN Brain Science Institute, Wako-shi, Saitama, JAPAN
cia@brain.riken.jp

Abstract. In the paper we derive and discuss a wide class of algorithms for 3D Super-symmetric nonnegative Tensor Factorization (SNTF) or nonnegative symmetric PARAFAC, and as a special case: Symmetric Nonnegative Matrix Factorization (SNMF) that have many potential applications, including multi-way clustering, feature extraction, multi-sensory or multi-dimensional data analysis, and nonnegative neural sparse coding. The main advantage of the derived algorithms is relatively low complexity and, in the case of multiplicative algorithms possibility for straightforward extension of the algorithms to L-order tensors factorization due to some nice symmetric property. We also propose also to use a wide class of cost functions such as Squared Euclidean, Kullback Leibler I-divergence, Alpha divergence and Beta divergence. Preliminary experimental results confirm the validity and good performance of some of these algorithms, especially when the data have sparse representations.

1 Introduction - Problem Formulation

Tensors (also known as n-way arrays or multidimensional arrays) are used in a variety of applications ranging from Neuroscience and psychometrics to chemometrics [1–8]. Non-negative Matrix Factorization (NMF), Non-negative Tensor Factorization (NTF) and parallel factor analysis (PARAFAC) models with non-negativity constraints have been recently proposed as sparse and quite efficient representations of signals, images, or general data [3, 4, 2, 5, 9–15]. From a viewpoint of data analysis, NTF is very attractive because it takes into account spatial and temporal correlations between variables more accurately than 2D matrix factorizations, such as NMF, and it usually provides sparse common factors or hidden (latent) components with physiological meaning and interpretation [5]. In most applications, especially in Neuroscience (EEG, fMRI), the PARAFAC models have been used [12, 16, 17]. In this paper, we consider the special form of the PARAFAC model (referred to here as the SNTF model), but with additional nonnegativity and sparsity constraints [6, 7, 18, 19]. In general case, the PARAFAC model can be described as a factorization of a given

* Also with IBS PAN and Warsaw University of Technology, Dept. of EXE, Warsaw, POLAND

** Also with Institute of Telecommunications, Information and Acoustics, Wroclaw University of Technology, POLAND

3D tensor $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I \times J \times Q}$ into three unknown matrices: $\boldsymbol{A} \in \mathbb{R}^{I \times J}$ representing the common factors, basis matrix, dictionary matrix or mixing matrix (depending on the applications), $\boldsymbol{D} \in \mathbb{R}^{Q \times J}$ usually representing scaling matrix, and $\boldsymbol{X} \in \mathbb{R}^{J \times T}$ representing second common factors, hidden components or sources (See Fig.1).

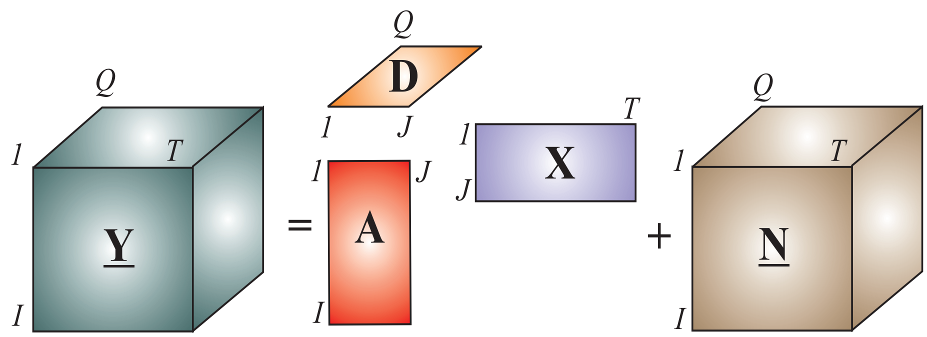

Fig. 1. General 3D PARAFAC model described by the set of matrix equations $\boldsymbol{Y}_q = \boldsymbol{A}\boldsymbol{D}_q\boldsymbol{X} + \boldsymbol{N}_q$, $q = 1, 2, \ldots, Q$, where $\boldsymbol{D}_q$ is a diagonal matrix that holds on the main diagonal the q-th row of $\boldsymbol{D}$. In the special case, of the SNTF, we impose nonnegativity and additional constraints that $I = T = Q$ and $\boldsymbol{A} = \boldsymbol{D} = \boldsymbol{G} \in \mathbb{R}^{I \times J}$ and $\boldsymbol{X} = \boldsymbol{G}^T$.

A super-symmetric tensor is a tensor whose entries are invariant under any permutation of the indices. For example, a third order super-symmetric tensor $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I \times T \times Q}$ (with $I = T = Q$) has $y_{itq} = y_{iqt} = y_{tiq} = y_{tqi} = y_{qit} = y_{qti}$.

Super-symmetric tensors arise naturally in multi-way clustering where they represent generalized affinity tensors, in higher order statistics, and blind source separation. Pierre Comon [20] has shown a nice relationship between super-symmetric tensors and polynomials. Zass and Shashua applied them to multi-way clustering problems [6, 19, 7], and Hazan et al. developed some multiplicative algorithms for the NTF [2].

We formulate the SNTF decomposition of a third order super-symmetric tensor $\underline{\boldsymbol{Y}} \in \mathbb{R}^{I \times I \times I}$ as three identical sparse nonnegative matrices $\boldsymbol{G} = [\boldsymbol{g}_1, \boldsymbol{g}_2, \ldots, \boldsymbol{g}_J] \in \mathbb{R}^{I \times J}$ with $J << I$ according to the following factorization:

$$\underline{\boldsymbol{Y}} = \sum_{j=1}^{J} (\boldsymbol{g}_j \circ \boldsymbol{g}_j \circ \boldsymbol{g}_j) + \underline{\boldsymbol{N}}, \qquad (1)$$

where $\boldsymbol{g}_j \in \mathbb{R}^I$ is the j-th column vector of the matrix $\boldsymbol{G}$, the operator $\circ$ means outer product[1], and $\underline{\boldsymbol{N}}$ is a tensor representing error.

The SNTF model can be described in the equivalent matrix form as

$$\boldsymbol{Y}_q = \boldsymbol{G}\boldsymbol{D}_q\boldsymbol{G}^T + \boldsymbol{N}_q, \qquad (q = 1, 2, \ldots, I) \qquad (2)$$

[1] Note that if $\boldsymbol{u}, \boldsymbol{v}, \boldsymbol{w}$ are vectors, then $[\boldsymbol{u} \circ \boldsymbol{v} \circ \boldsymbol{w}]_{ijq} = u_i v_j w_q$.

where $\boldsymbol{Y}_q = \underline{\boldsymbol{Y}}_{:,:,q} = [y_{itq}] \in \mathbb{R}_+^{I \times I}$ are (frontal) slices of the given tensor $\underline{\boldsymbol{Y}} \in \mathbb{R}_+^{I \times I \times I}$, $I = Q = T$ is the number of the (horizontal, vertical, frontal) slices, $\boldsymbol{G} = [g_{ij}]_{I \times J} \in \mathbb{R}_+^{I \times J}$ is the unknown matrix (super-common factor) to be estimated, $\boldsymbol{D}_q \in \mathbb{R}_+^{J \times J}$ is a diagonal matrix that holds the q-th row of $\boldsymbol{G}$ in its main diagonal, and $\boldsymbol{N}_q = \underline{\boldsymbol{N}}_{:,:,q} \in \mathbb{R}^{I \times I}$ is the q-th frontal slice of a tensor $\underline{\boldsymbol{N}} \in \mathbb{R}^{I \times I \times I}$ (not necessary super symmetric) representing error or noise, depending upon the application. The above algebraic system can be represented in an equivalent scalar form as follows

$$y_{itq} = z_{itq} + n_{itq} = \sum_{j=1}^{J} g_{ij}\ g_{tj}\ g_{qj} + n_{itq}. \tag{3}$$

The objective is to estimate a sparse matrix $\boldsymbol{G}$, subject to some constraints like scaling to unit length vectors, non-negativity and other possible natural constraints such as orthogonality, sparseness and/or smoothness of all or some of the columns $\boldsymbol{g}_j$.

Throughout this paper, we use the following notation: the ij-th element of the matrix $\boldsymbol{G}$ is denoted by g_{ij}, and its j-th column by $\boldsymbol{g}_j$, $y_{itq} = [\boldsymbol{Y}_q]_{it}$ means the it-th element of the q-th frontal slice $\boldsymbol{Y}_q$ and $z_{itq} = \sum_{j=1}^{J} g_{ij} g_{tj} g_{qj}$ with $(i = 1, 2, \ldots, I;\ t = 1, 2, \ldots, I;\ q = 1, 2, \ldots, I)$.

2 Multiplicative SNTF algorithms

2.1 Generalized Alpha Divergence

The most widely known and often used adaptive algorithms for NTF/NMF and also SNTF are based on alternating minimization of the squared Euclidean distance and the generalized Kullback-Leibler divergence [15, 13, 9]. In this paper, we propose to use a more general cost function: Alpha divergence.

The 3D generalized alpha- divergence can be defined for our purpose as follows [1]:

$$D_A^{(\alpha)}(\underline{\boldsymbol{Y}}||\underline{\boldsymbol{Z}}) = \begin{cases} \displaystyle\sum_{itq} \left(\frac{y_{itq}}{\alpha(\alpha-1)} \left[\left(\frac{y_{itq}}{z_{itq}} \right)^{\alpha-1} - 1 \right] - \frac{y_{itq} - z_{itq}}{\alpha} \right), & \alpha \neq 0, 1, \\ \displaystyle\sum_{itq} \left(y_{itq}\ \ln \left(\frac{y_{itq}}{z_{itq}} \right) - y_{itq} + z_{itq} \right) & \alpha = 1, \\ \displaystyle\sum_{itq} \left(z_{itq} \ln \left(\frac{z_{itq}}{y_{itq}} \right) + y_{itq} - z_{itq} \right), & \alpha = 0, \end{cases} \tag{4}$$

where $y_{itq} = [\underline{\boldsymbol{Y}}]_{itq}$ and $z_{itq} = [\boldsymbol{G}\boldsymbol{D}_q\boldsymbol{G}^T]_{it}$ for $(i = 1, 2, \ldots, I),\ (t = 1, 2, \ldots, I),\ (q = 1, 2, \ldots, I)$.

The choice of the parameter $\alpha \in \mathbb{R}$ depends on the statistical distribution of noise and data. We recall, that as special cases of the alpha-divergence for $\alpha = 2, 0.5, -1$, we obtain the Pearson's chi squared, Hellinger and Neyman's chi-square distances, respectively, while for the cases $\alpha = 1$ and $\alpha = 0$ the divergence has to be defined by the limits of (4 (a)) as $\alpha \to 1$ and $\alpha \to 0$, respectively. When these limits are evaluated, one obtains the generalized Kullback-Leibler divergence defined by equations (4 (c)) for $\alpha \to 1$.

The gradient of the alpha divergence (4), for $\alpha \neq 0$, can be expressed in compact form as:

$$\frac{\partial D_A^{(\alpha)}}{\partial g_{ij}} = \frac{1}{\alpha} \sum_{tq} g_{qj}\, g_{tj} \left[1 - \left(\frac{y_{itq}}{z_{itq}} \right)^{\alpha} \right], \qquad \alpha \neq 0. \tag{5}$$

However, instead of applying here the standard gradient descent, we use a projected (nonlinearly transformed) gradient approach (which can be considered as a generalization of the exponentiated gradient):

$$\Phi(g_{ij}) \leftarrow \Phi(g_{ij}) - \eta_{ij} \frac{\partial D_A^{(\alpha)}}{\partial g_{ij}}, \tag{6}$$

where $\Phi(x)$ is a suitable chosen function.

Hence, we have

$$g_{ij} \leftarrow \Phi^{-1} \left(\Phi(g_{ij}) - \eta_{ij} \frac{\partial D_A^{(\alpha)}}{\partial g_{ij}} \right), \tag{7}$$

It can be shown that using such nonlinear scaling or transformation provides a stable solution and the gradients are much better behaved in the Φ space. In our case, we employ $\Phi(x) = x^{\alpha}$ and choose the learning rates: $\eta_{ij} = \alpha \Phi(g_{ij}) / \sum_{tq} g_{qj}\, g_{jt}$, which leads to the generalized multiplicative alpha algorithm [2]:

$$g_{ij} \leftarrow g_{ij} \left(\frac{\sum_{tq} g_{qj}\, g_{jt}\; (y_{itq}/z_{itq})^{\alpha}}{\sum_{tq} g_{qj}\, g_{tj}} \right)^{1/\alpha}, \tag{8}$$

with the normalization of the columns of $\boldsymbol{G}$ to unit length at each iteration, i.e.: $g_{ij} \leftarrow g_{ij} / \sum_{p=1}^{I} g_{pj}$. This SNTF algorithm can be considered as a generalization of the EMML algorithm (for $\alpha = 1$) proposed in [2, 6].

We may apply nonlinear projections or filtering via suitable nonlinear monotonic functions which increase or decrease the sparseness. In the simplest case, we can apply a very simple nonlinear transformation $g_{tj} \leftarrow (g_{tj})^{1+\alpha_{sp}}$, $\forall k$, where α_{sp} is a small coefficient, typically from 0.001 to 0.005, and it is positive if we want to increase sparseness of an estimated component and negative if we want to decrease the sparseness. Hence, the generalized alpha algorithm for the SNTF

[2] For $\alpha = 0$, instead of $\Phi(x) = x^{\alpha}$ we have used $\Phi(x) = \ln(x)$.

with sparsity control can take the following form:

$$g_{ij} \leftarrow \left[g_{ij} \left(\frac{\sum_{tq} g_{qj}\, g_{tj}\ (y_{itq}/z_{itq})^{\alpha}}{\sum_{tq} g_{qj}\, g_{tj}} \right)^{\omega/\alpha} \right]^{1+\alpha_{sp}}, \tag{9}$$

where ω is an over-relaxation parameter (typically, in the range $(0, 2)$) which controls the convergence speed, and α_{sp} is a small parameter which controls sparsity of the estimated matrix $\boldsymbol{G}$.

2.2 SMART Algorithm

Alternative multiplicative SNTF algorithms can be derived using the exponentiated gradient (EG) descent updates instead of the standard additive gradient descent. For example, by using the alpha divergence (4) for $\alpha = 0$, we have

$$g_{ij} \leftarrow g_{ij} \exp\left(-\tilde{\eta}_i \frac{\partial D_A^{(0)}}{\partial g_{ij}} \right), \tag{10}$$

$$\frac{\partial D_A^{(0)}}{\partial g_{ij}} = \sum_{tq} g_{qj}\ g_{tj}\, (\ln z_{itq} - \ln y_{itq})\,. \tag{11}$$

Hence, we obtain the simple multiplicative learning rules:

$$g_{ij} \leftarrow g_{ij} \exp\left(\sum_{tq} \eta_{ij} g_{qj} g_{tj} \ln\left(\frac{y_{itq}}{z_{itq}} \right) \right) = g_{ij} \prod_{tq} \left(\frac{y_{itq}}{z_{itq}} \right)^{\eta_{ij}\, g_{qj}\, g_{tj}}. \tag{12}$$

The nonnegative learning rates η_{ij} can take different forms. Typically, in order to guarantee stability of the algorithm we assume that $\eta_{ij} = \tilde{\eta}_j = \omega\, (\sum_{t=1}^{T} g_{tj})^{-2}$, where $\omega \in (0, 2)$ is an over-relaxation parameter.

The above SNTF multiplicative algorithm can be considered as an alternating minimization/projection extension of the well-known SMART (Simultaneous Multiplicative Algebraic Reconstruction Technique) [11, 21].

2.3 Generalized Beta Divergence

The generalized beta divergence can be considered as complementary cost function to the generalized alpha divergence and can be defined as follows:

$$D_B^{(\beta)}(\underline{\boldsymbol{Y}}||\underline{\boldsymbol{Z}}) = \begin{cases} \sum_{itq} \left(y_{itq} \dfrac{y_{itq}^{\beta} - z_{itq}^{\beta}}{\beta} - \dfrac{y_{itq}^{\beta+1} - z_{itq}^{\beta+1}}{\beta+1} \right), & \beta > 0, \\ \sum_{itq} \left(y_{itq}\ \ln(\dfrac{y_{itq}}{z_{itq}}) + y_{itq} - z_{itq} \right) & \beta = 0, \\ \sum_{itq} \left(\ln(\dfrac{z_{itq}}{y_{itq}}) + \dfrac{y_{itq}}{z_{itq}} - 1 \right), & \beta = -1. \end{cases} \tag{13}$$

The choice of the parameter β depends on statistical distribution of the data and the beta divergence corresponds to the Tweedie models [22]. For example, the optimal choice of the parameter β for a normal distribution is $\beta = 1$, for a gamma distribution is $\beta = -1$, for a Poisson distribution is $\beta = 0$, and for the compound Poisson $\beta \in (-1, 0)$.

From the beta generalized divergence, we can derive various kinds of SNTF algorithms: Multiplicative algorithms based on the standard gradient descent or the Exponentiated Gradient (EG) algorithms, additive algorithms using Projected Gradient (PG) or Interior Point Gradient (IPG), quasi-Newton and Fixed Point (FP) ALS algorithms [23–28, 9, 13].

In order to derive the multiplicative SNTF learning algorithm for a sparse factorization, we compute the gradient of a regularized beta divergence (13), with the additional regularization (sparsification) term $J(\boldsymbol{G}) = \alpha_G ||\boldsymbol{G}||_1 = \alpha_G \sum_{ij} g_{ij}$ as:

$$\frac{\partial D_{Breg}^{(\beta)}}{\partial g_{ij}} = \sum_{tq} \left(z_{itq}^{\beta} - y_{itq} z_{itq}^{\beta-1} \right) \; g_{qj} \, g_{tj} + \alpha_G. \tag{14}$$

Applying the simple (the first-order) gradient descent approach:

$$g_{ij} \leftarrow g_{ij} - \eta_{ij} \frac{\partial D_{Breg}^{(\beta)}}{\partial g_{ij}}, \tag{15}$$

and by choosing suitable learning rates: $\eta_{ij} = g_{ij} / \sum_{tq} z_{itq}^{\beta} \, g_{qj} \, g_{tj}$, we obtain a generalized SNTF beta algorithm:

$$g_{ij} \leftarrow g_{ij} \; \frac{\left[\displaystyle\sum_{tq} g_{qj} \, g_{tj} (y_{jtq} / z_{itq}^{1-\beta}) - \alpha_G \right]_+}{\sum_{tq} z_{itq}^{\beta} \; g_{qj} \, g_{tj}}, \tag{16}$$

where $[x]_{\varepsilon} = \max\{\varepsilon, x\}$ with a small $\varepsilon = 10^{-16}$ introduced to avoid zero and negative values.

In the special case, for $\beta = 0$ the above algorithm simplifies to the generalized alternating EMML algorithm that is similar to the algorithm derived by Hazan et al. [2, 29]:

$$g_{ij} \leftarrow g_{ij} \; \frac{\left[\displaystyle\sum_{tq} g_{qj} \, g_{tj} (y_{jtq} / z_{itq}) - \alpha_G \right]_+}{\sum_{tq} g_{qj} \, g_{tj}}. \tag{17}$$

3 Simple Alternative Approaches for Super-Symmetric Tensor Decomposition

3.1 Averaging Approach

For large dimensions of tensors ($I >> 1$), the above derived local algorithm could be computationally very time consuming.

In this section, we propose an alternative simple approach which converts the problem to a simple tri-NMF model:

$$\bar{\boldsymbol{Y}} = \boldsymbol{G}\bar{\boldsymbol{D}}\boldsymbol{G}^T + \bar{\boldsymbol{N}}, \tag{18}$$

where $\bar{\boldsymbol{Y}} = \sum_{q=1}^{Q} \boldsymbol{Y}_q \in \mathbb{R}^{I\times I}$, $\bar{\boldsymbol{D}} = \sum_{q=1}^{Q} \boldsymbol{D}_q = \mathrm{diag}\{\bar{d}_1, \bar{d}_2, \ldots, \bar{d}_J\}$ and $\bar{\boldsymbol{N}} = \sum_{q=1}^{Q} \boldsymbol{N}_q \in \mathbb{R}^{I\times I}$.

The above system of linear algebraic equations can be represented in an equivalent scalar form as: $\bar{y}_{it} = \sum_j g_{ij} g_{tj}\, \bar{d}_j + \bar{n}_{it}$ or equivalently in the vector form: $\bar{\boldsymbol{Y}} = \sum_j \boldsymbol{g}_j\ \bar{d}_j\ \boldsymbol{g}_j^T + \bar{\boldsymbol{N}}$ where $\boldsymbol{g}_j$ are columns of $\boldsymbol{G}$.

Such a simple model is justified if noise in the frontal slices is uncorrelated. It is interesting to note that the model can be written in the equivalent form:

$$\bar{\boldsymbol{Y}} = \tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^T + \bar{\boldsymbol{N}}, \tag{19}$$

where $\tilde{\boldsymbol{G}} = \boldsymbol{G}\bar{\boldsymbol{D}}^{1/2}$, assuming that $\bar{\boldsymbol{D}} \in \mathbb{R}^{J\times J}$ is a non-singular matrix. Thus, the problem can be converted to a standard symmetric NMF problem to estimate matrix $\tilde{\boldsymbol{G}}$. Using any available NMF algorithm: Multiplicative, FP-ALS, or PG, we can estimate the matrix $\tilde{\boldsymbol{G}}$.

For example, by minimizing the following regularized cost function:

$$D(\bar{\boldsymbol{Y}}||\tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^T) = \frac{1}{2}||\bar{\boldsymbol{Y}} - \tilde{\boldsymbol{G}}\tilde{\boldsymbol{G}}^T||_F^2 + \alpha_G||\tilde{\boldsymbol{G}}||_1 \tag{20}$$

and applying the FP-ALS approach, we obtain the following simple algorithm

$$\tilde{\boldsymbol{G}} \leftarrow \left[(\bar{\boldsymbol{Y}}^T\tilde{\boldsymbol{G}} - \alpha_G\ \boldsymbol{E})(\tilde{\boldsymbol{G}}^T\tilde{\boldsymbol{G}})^{-1}\right]_+, \tag{21}$$

subject to normalization the columns of $\tilde{\boldsymbol{G}}$ to unit-length in each iteration step, where $\boldsymbol{E}$ is the matrix of all ones of appropriate size.

3.2 Row-wise and Column-wise Unfolding Approach

It is worth noting that the diagonal matrices $\boldsymbol{D}_q$ are scaling matrices that can be absorbed by the matrix $\boldsymbol{G}$. By defining the column-normalized matrices $\boldsymbol{G}_q = \boldsymbol{G}\boldsymbol{D}_q$, we can use the following simplified models:

$$\boldsymbol{Y}_q = \boldsymbol{G}_q\boldsymbol{G}^T + \boldsymbol{N}_q, \qquad (q = 1, \ldots, Q) \tag{22}$$

or equivalently

$$\boldsymbol{Y}_q = \boldsymbol{G}\boldsymbol{G}_q^T + \boldsymbol{N}_q, \qquad (q = 1, \ldots, Q). \tag{23}$$

These simplified models can be described by a single compact matrix equation using column-wise or row-wise unfolding as follows

$$\boldsymbol{Y}_c = \boldsymbol{G}_c\boldsymbol{G}^T, \tag{24}$$

or

$$\boldsymbol{Y}_r = \boldsymbol{G}\boldsymbol{G}_r^T, \tag{25}$$

where $\boldsymbol{Y}_c = \boldsymbol{Y}_r^T = [\boldsymbol{Y}_1; \boldsymbol{Y}_2; \ldots; \boldsymbol{Y}_Q] \in \mathbb{R}^{I^2 \times I}$ is the column-wise unfolded matrix of the slices $\boldsymbol{Y}_q$ and $\boldsymbol{G}_c = \boldsymbol{G}_r^T = [\boldsymbol{G}_1; \boldsymbol{G}_2; \ldots; \boldsymbol{G}_Q] \in \mathbb{R}^{JI \times I}$ is column-wise unfolded matrix of the matrices $\boldsymbol{G}_q = \boldsymbol{G}\boldsymbol{D}_q$ $(q = 1, 2, , \ldots, I)$.

Using any efficient NMF algorithm (multiplicative, IPN, quasi-Newton, or FP-ALS) [23–28, 9, 13], we can estimate the matrix $\boldsymbol{G}$. For example, by minimizing the following cost function:

$$D(\boldsymbol{Y}_c||\boldsymbol{G}_c\boldsymbol{G}^T) = \frac{1}{2}||\boldsymbol{Y}_c - \boldsymbol{G}_c\boldsymbol{G}^T||_F^2 + \alpha_G\, ||\boldsymbol{G}||_1 \tag{26}$$

and applying the FP-ALS approach, we obtain the following iterative algorithm:

$$\boldsymbol{G} \leftarrow \left[([\boldsymbol{Y}_c^T\boldsymbol{G}_c - \alpha_G \boldsymbol{E}]_+)(\boldsymbol{G}_c^T\boldsymbol{G}_c)^{-1}\right]_+ \tag{27}$$

or equivalently

$$\boldsymbol{G} \leftarrow \left[([\boldsymbol{Y}_r\boldsymbol{G}_r - \alpha_G\, \boldsymbol{E}]_+)(\boldsymbol{G}_r^T\boldsymbol{G}_r)^{-1}\right]_+, \tag{28}$$

where $\boldsymbol{G}_c = \boldsymbol{G}_r^T = [\boldsymbol{G}\boldsymbol{D}_1; \boldsymbol{G}\boldsymbol{D}_2; \ldots; \boldsymbol{G}\boldsymbol{D}_Q]$, $\boldsymbol{D}_q = \mathrm{diag}\{\underline{\boldsymbol{g}}_q\}$ and $\underline{\boldsymbol{g}}_q$ means q-th row of $\boldsymbol{G}$.

3.3 Semi-orthogonality Constraint

The matrix $\boldsymbol{G}$ is usually very sparse and additionally satisfies orthogonality constraints. We can easily impose orthogonality constraint by incorporating additionally the following iterations:

$$\boldsymbol{G} \leftarrow \boldsymbol{G}\left(\boldsymbol{G}^T\boldsymbol{G}\right)^{-1/2}. \tag{29}$$

3.4 Simulation Results

All the NTF algorithms presented in this paper have been tested for many difficult benchmarks for signals and images with various statistical distributions of signals and additive noise. Comparison and simulation results will be presented in the ICONIP-2007.

4 Conclusions and Discussion

We have proposed the generalized and flexible cost function (controlled by sparsity penalty/regularization terms) that allows us to derive a family of SNTF algorithms. The main objective and motivations of this paper is to derive simple multiplicative algorithms which are especially suitable both for very sparse representation and highly over-determined cases. The basic advantage of the multiplicative algorithms is their simplicity and relatively straightforward generalization to L-order tensors ($L > 3$). However, the multiplicative algorithms are relatively slow.

We found that simple approaches which convert a SNTF problem to a symmetric NMF (SNMF) or symmetric tri-NMF (ST-NMF) problem provide the more efficient and fast algorithms, especially for large scale problems. Moreover, by imposing orthogonality constraints, we can drastically improve performance, especially for noisy data.

Obviously, there are many challenging open issues remaining, such as global convergence and an optimal choice of the associated parameters.

References

1. Amari, S.: Differential-Geometrical Methods in Statistics. Springer Verlag (1985)
2. Hazan, T., Polak, S., Shashua, A.: Sparse image coding using a 3D non-negative tensor factorization. In: International Conference of Computer Vision (ICCV). (2005) 50–57
3. : Workshop on tensor decompositions and applications, CIRM, Marseille, France (2005)
4. Heiler, M., Schnoerr, C.: Controlling sparseness in non-negative tensor factorization. Springer LNCS **3951** (2006) 56–67
5. Smilde, A., Bro, R., Geladi, P.: Multi-way Analysis: Applications in the Chemical Sciences. John Wiley and Sons, New York (2004)
6. Shashua, A., Zass, R., Hazan, T.: Multi-way clustering using super-symmetric non-negative tensor factorization. In: European Conference on Computer Vision (ECCV), Graz, Austria (2006)
7. Zass, R., Shashua, A.: A unifying approach to hard and probabilistic clustering. In: International Conference on Computer Vision (ICCV), Beijing, China (2005)
8. Sun, J., Tao, D., Faloutsos, C.: Beyond streams and graphs: dynamic tensor analysis. In: Proc.of the 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining. (2006)
9. Berry, M., Browne, M., Langville, A., Pauca, P., Plemmons, R.: Algorithms and applications for approximate nonnegative matrix factorization. Computational Statistics and Data Analysis (2006) in press.
10. Cichocki, A., Zdunek, R., Amari, S.: Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. Springer LNCS **3889** (2006) 32–39
11. Cichocki, A., Amari, S., Zdunek, R., Kompass, R., Hori, G., He, Z.: Extended SMART algorithms for non-negative matrix factorization. Springer LNAI **4029** (2006) 548–562
12. Cichocki, A., Zdunek, R.: NTFLAB for Signal Processing. Technical report, Laboratory for Advanced Brain Signal Processing, BSI, RIKEN, Saitama, Japan (2006)
13. Dhillon, I., Sra, S.: Generalized nonnegative matrix approximations with Bregman divergences. In: Neural Information Proc. Systems, Vancouver, Canada (2005) 283–290
14. Kim, M., Choi, S.: Monaural music source separation: Nonnegativity, sparseness, and shift-invariance. Springer LNCS **3889** (2006) 617–624
15. Lee, D.D., Seung, H.S.: Learning the parts of objects by nonnegative matrix factorization. Nature **401** (1999) 788–791

16. Mørup, M., Hansen, L.K., Herrmann, C.S., Parnas, J., Arnfred, S.M.: Parallel factor analysis as an exploratory tool for wavelet transformed event-related EEG. NeuroImage **29** (2006) 938–947
17. Miwakeichi, F., Martinez-Montes, E., Valds-Sosa, P., Nishiyama, N., Mizuhara, H., Yamaguchi, Y.: Decomposing EEG data into space−time−frequency components using parallel factor analysi. NeuroImage **22** (2004) 1035–1045
18. Zass, R., Shashua, A.: Nonnegative sparse pca. In: Neural Information Processing Systems (NIPS), Vancuver, Canada (2006)
19. Zass, R., Shashua, A.: Doubly stochastic normalization for spectral clustering. In: Neural Information Processing Systems (NIPS), Vancuver, Canada (2006)
20. Comon, P.: Tensor decompositions-state of the art and applications. In McWhirter, J.G., Proudler, I.K., eds.: Institute of Mathematics and its Applications Conference on Mathematics in Signal Processing, Oxford, UK, Clarendon Press (2001) 18–20
21. Byrne, C.L.: Choosing parameters in block-iterative or ordered-subset reconstruction algorithms. IEEE Transactions on Image Processing **14** (2005) 321–327
22. Minami, M., Eguchi, S.: Robust blind source separation by Beta-divergence. Neural Computation **14** (2002) 1859–1886
23. Cichocki, A., Zdunek, R., Choi, S., Plemmons, R., Amari, S.I.: Novel multi-layer nonnegative tensor factorization with sparsity constraints. Springer LNCS **4432** (2007) 271–280
24. Cichocki, A., Zdunek, R.: Regularized alternating least squares algorithms for non-negative matrix/tensor factorizations. Springer LNCS **4493** (2007) 793–802
25. Cichocki, A., Zdunek, R., Amari, S.: New algorithms for non-negative matrix factorization in applications to blind source separation. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP2006. Volume 5., Toulouse, France (2006) 621–624
26. Cichocki, A., Zdunek, R., Choi, S., Plemmons, R., Amari, S.: Nonnegative tensor factorization using Alpha and Beta divergencies. In: Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP07). Volume III., Honolulu, Hawaii, USA (2007) 1393–1396
27. Zdunek, R., Cichocki, A.: Nonnegative matrix factorization with constrained second-order optimization. Signal Processing **87** (2007) 1904–1916
28. Zdunek, R., Cichocki, A.: Nonnegative matrix factorization with quadratic programming. Neurocomputing (2007) accepted.
29. Shashua, A., Hazan, T.: Non-negative tensor factorization with applications to statistics and computer vision. In: Proc. of the 22-th International Conference on Machine Learning, Bonn, Germany (2005)