Algorithms for Sparse Non-negative Tucker decompositions

Morten Mørup and Lars Kai Hansen
Technical University of Denmark, 2800 Kongens Lyngby, Denmark

Sidse M. Arnfred
Department of Psychiatry, Hvidovre hospital, University Hospital of Copenhagen, Denmark

Abstract There is a increasing interest in analysis of large scale multi-way data. The concept of multi-way data refers to arrays of data with more than two dimensions, i.e., taking the form of tensors. To analyze such data, decomposition techniques are widely used. The two most common decompositions for tensors are the Tucker model and the more restricted PARAFAC model. Both models can be viewed as generalizations of the regular factor analysis to data of more than two modalities. Non-negative matrix factorization (NMF) in conjunction with sparse coding has lately been given much attention due to its part based and easy interpretable representation. While NMF has been extended to the PARAFAC model no such attempt has been done to extend NMF to the Tucker model. However, if the tensor data analyzed is non-negative it may well be relevant to consider purely additive (i.e., non-negative Tucker decompositions). To reduce ambiguities of this type of decomposition we develop updates that can impose sparseness in any combination of modalities, hence, proposed algorithms for sparse non-negative Tucker decompositions (SN-TUCKER). We demonstrate how the proposed algorithms are superior to existing algorithms for Tucker decompositions when indeed the data and interactions can be considered non-negative. We further illustrate how sparse coding can help identify what model (PARAFAC or Tucker) is the most appropriate for the data as well as to select the number of components by turning off excess components. The algorithms for SN-

TUCKER can be downloaded from [Mørup, 2007].

1 Introduction

Tensor decompositions are in frequent use today in a variety of fields including psychometric, chemometrics, image analysis, graph analysis and signal processing [Murakami and Kroonenberg, 2003; Vasilescu and Terzopoulos, 2002; Wang and Ahuja, 2003; Jia and Gong, 2005; Sun et al., 2005; Gurden et al., 2001; Nørgaard and Ridder, 1994; Smilde et al., 1999, 2004; Andersson and Bro, 1998]. Tensors, i.e., $\mathcal{X} \in \Re^{I_1 \times I_2 \times \ldots \times I_N}$, also called multi-way arrays or multidimensional matrices are generalizations of vectors (first order tensors) and matrices (second order tensors). The two most commonly used decompositions of tensors are the Tucker model [Tucker, 1966] and the more restricted PARAFAC/CANDECOMP model [Harshman, 1970; Carroll and Chang, 1970].

The Tucker model reads

$$\mathcal{X}_{i_1,i_2,\ldots,i_N} \approx \mathcal{R}_{i_1,i_2,\ldots,i_N} = \sum_{j_1 j_2 \ldots j_N} \mathcal{G}_{j_1,j_2,\ldots,j_N} \mathbf{A}^{(1)}_{i_1,j_1} \mathbf{A}^{(2)}_{i_2,j_2} \cdot \ldots \cdot \mathbf{A}^{(N)}_{i_N,j_N}. \quad (1)$$

where $\mathcal{G} \in \Re^{J_1 \times J_2 \times \ldots \times J_N}$ and $\mathbf{A}^{(n)} \in \Re^{I_n \times J_n}$. To indicate how many vectors pertain to each modality it is customary also to denote the model a Tucker $J_1 - J_2 - \cdots - J_N$. Using the n-mode tensor product $\times_n$ [Lathauwer et al., 2000] given by

$$(\mathcal{Q} \times_n \mathbf{P})_{i_1,i_2,\ldots,j_n,\ldots i_N} = \sum_{i_n} \mathcal{Q}_{i_1,i_2,\ldots,i_n,\ldots i_N} \mathbf{P}_{j_n,i_n}, \quad (2)$$

the model is stated as

$$\mathcal{X} \approx \mathcal{R} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 \ldots \times_N \mathbf{A}^{(N)}. \quad (3)$$

The Tucker model represents the data spanning the n^{th} modality by the vectors (loadings) given by the J_n columns of $\mathbf{A}^{(n)}$ such that the vectors of each modality interact with the vectors of all remaining modalities with strengths given by a so-called core tensor $\mathcal{G}$. As a result, the Tucker model encompass all possible linear interactions between vectors pertaining to the various modalities of the data.

The PARAFAC model is a special case of the Tucker model where the size of each modality of the core array $\mathcal{G}$ is the same, i.e., $J_1 = J_2 = \cdots = J_N$ while interaction is only between columns of same indices such that the only non-zero elements are found along the diagonal of the core, i.e., $\mathcal{G}_{j_1,j_2,\ldots,j_N} \neq 0$ iff $j_1 = j_2 = \ldots = j_N$. Notice, in the Tucker model a rotation of a given loading matrix $\mathbf{A}^{(n)}$ can be compensated by a counter rotation of the core $\mathcal{G}$, i.e., $\mathcal{G} \times_n \mathbf{A}^{(n)} = (\mathcal{G} \times_n \mathbf{P}^{-1}) \times_n (\mathbf{A}^{(n)}\mathbf{P})$. While the factors of the unconstrained Tucker model are orthogonal, this is not the case for the factors of the PARAFAC model. Furthermore, as the PARAFAC model requires the core to be diagonal this restricts $\mathbf{P}$ in general to be a simple scale and permutation matrix. Thus, contrary to the PARAFAC model [Kruskal, 1977; Sidiropoulos and Bro, 2000] the Tucker model is not unique in general.

Non-negative matrix factorization (NMF) is given by the decomposition

$$\mathbf{V} \approx \mathbf{R} = \mathbf{WH}, \tag{4}$$

where $\mathbf{V} \in \Re_+^{N \times M}$, $\mathbf{W} \in \Re_+^{N \times D}$ and $\mathbf{H} \in \Re_+^{D \times M}$, i.e., such that the variables $\mathbf{V}$, $\mathbf{W}$ and $\mathbf{H}$ are non-negative. The decomposition is useful as it results in easy interpretable part based representations [Lee and Seung, 1999]. Non-negative decomposition is also named positive matrix factorization [Paatero and Tapper, 1994] but was popularized by Lee and Seung [1999, 2000] due to a simple and efficient algorithmic procedure based on multiplicative updates. The decomposition has proven useful for a wide range of data where non-negativity is a natural constraint. These encompass data for text-mining based on word frequencies, image data, biomedical data and spectral data. The algorithm can even be useful when the data inherently is indefinite, but after transformation becomes non-negative, say audio, where NMF has been successfully used for analysis of the amplitude of a spectral representation [Smaragdis and Brown, 2003].

Unfortunately, the decomposition is not in general unique [Donoho and Stodden, 2003]. However, sparseness has been imposed such that ambiguities are reduced by finding the solution being the most sparse (by some measure of sparseness). This is often also the most simple, i.e., parsimonious solution to the data [Olshausen and Field, 2004; Eggert and Körner, 2004; Hoyer, 2004]. Non-negative matrix factor-

ization has recently been extended to the PARAFAC model [Welling and Weber, 2001; FitzGerald et al., 2005; Parry and Essa, 2006; Cichocki et al., 2007]. However, despite the attractive properties of non-negative decompositions and sparse coding neither approaches have so far been extended to the Tucker model.

Traditionally, the Tucker model has been estimated using various alternating least squares algorithms where the columns of $\mathbf{A}^{(n)}$ for the unconstrained Tucker are orthogonal [Andersson and Bro, 1998]. Recently, an algorithm for higher order singular value decomposition (HOSVD) based on solving N eigenvalue problems to estimate the Tucker model was given [Lathauwer et al., 2000]. However, just as NMF does not have orthogonal factors neither will factors in the constrained Tucker model be forced orthogonal. Although algorithms for non-negative Tucker decompositions exist [Bro and Andersson, 2000] the decompositions do not allow for the core to be constrained non-negative. Furthermore, the decompositions are in general ambiguous. Consequently, the lack of uniqueness hampers interpretability of these decompositions. For this reason the existing non-negative Tucker decompositions have not been widely used. Presently, we will develop multiplicative algorithms for fully non-negative Tucker decompositions, i.e., forming a non-negative Tucker decomposition where both data, core and loadings are non-negative. Ambiguities of the decompositions are reduced imposing sparseness such that the solution being the sparsest according to some measure of sparsity is attained.

In the following $\mathcal{X}_b^a$ will denote a tensor of the modalities a containing data of type b. Recently, the Tucker model has among others been applied to:

1. Spectroscopy data ([Smilde et al., 2004; Andersson and Bro, 1998] for instance $\mathcal{X}_{Strength}^{Batch\ number \times Time \times Spectra}$[Gurden et al., 2001; Nørgaard and Ridder, 1994; Smilde et al., 1999])
2. Web mining ($\mathcal{X}_{Click\ counts}^{Users \times Queries \times Wep\ pages}$ [Sun et al., 2005])
3. Image analysis ($\mathcal{X}_{Image\ intensity}^{People \times Views \times Illuminations \times Expressions \times Pixels}$ [Vasilescu and Terzopoulos, 2002; Wang and Ahuja, 2003; Jia and Gong, 2005]
4. Semantic differential data ($\mathcal{X}_{Grade}^{Judges \times Music\ pieces \times Scales}$[Murakami and

Kroonenberg, 2003])

All the above data sets are non-negative and the basis vectors/projections $\mathbf{A}^{(n)}$ and interactions $\mathcal{G}$ can be assumed additive, viz., non-negative. For the spectroscopy data non-negativity would yield batch groups containing, time and spectra profiles additively combined by the non-negative core, for the web mining data giving groups of users, queries and web pages interrelated with a strength given by the non-negative core etc. However, none of the Tucker analysis above have considered such purely non-negative decompositions where the "whole" is modeled as the sum of its "parts" resulting in easy interpretable part based representation.

The paper is structured as follows: First, two algorithms for sparse non-negative Tucker (SN-TUCKER) decomposition based on a gaussian noise model (i.e., least squares (LS) minimization) and Poisson noise (i.e., Kulback-Leibler (KL) divergence minimization) are derived. The derivation easily generalizes to other types of objective functions such as Bregman, Ciszar, α and β divergences [Dhillon and Sra, 2005; Cichocki et al., 2006, 2007], however, the focus is here on LS and KL, since they are the two most widely used objective functions for NMF. Next, the algorithms ability to identify the components of synthetically generated data is demonstrated. Finally, the algorithms are tested on two real data sets, one of wavelet transformed EEG previously explored by the PARAFAC model [Mørup et al., 2006] the other a data set obtained from a flow injection analysis (FIA) [Nørgaard and Ridder, 1994; Smilde et al., 1999]. The applications demonstrate different aspects of the SN-TUCKER model.

2 Methods

In the following $A \bullet B$ and $\frac{A}{B}$ will denote element-wise multiplication and division, respectively, while $(\mathbf{M})^{\cdot\alpha}$ denotes elements-wise raising the elements of $\mathbf{M}$ to the α^{th} power. $\mathcal{E}$, $\mathbf{E}$ and $\mathbf{1}$ will, respectively, denote a tensor, a matrix, and a vector of ones in all entries. Finally, $\bullet$ supersedes $\cdot$ where $\cdot$ denotes the regular matrix multiplication.

The sparse non-negative Tucker (SN-TUCKER) algorithms proposed here is based on the multiplicative updates introduced in [Lee

and Seung, 1999, 2000; Lee et al., 2002] for non-negative matrix factorization (NMF). Although, other types of updates exists for non-negativity constraint optimization such as projected gradient [Lin, 2007] and active sets [Bro and Jong, 1997], multiplicative updates are simple to implement and extend well to sparse coding [Eggert and Körner, 2004]. Consider the cost function $C(\theta)$ of the non-negative variables θ. Let further $\frac{\partial C(\theta)_i^+}{\partial \theta_i}$ and $\frac{\partial C(\theta)_i^-}{\partial \theta_i}$ be the positive and negative part of the derivative with respect to θ_i. Then the multiplicative update has the following form:

$$\theta_i \leftarrow \theta_i \left(\frac{\frac{\partial C(\theta)^-}{\partial \theta_i}}{\frac{\partial C(\theta)^+}{\partial \theta_i}} \right)^{\alpha}. \tag{5}$$

A small constant $\varepsilon = 10^{-9}$ can be added to the denominator to avoid potential division by zero. By also adding the constant to the numerator the corresponding gradient is unaltered. When the gradient is zero $\frac{\partial C(\theta)}{\partial \theta_i}^+ = \frac{\partial C(\theta)^-}{\partial \theta_i}$ such that θ is left unchanged. If the gradient is positive $\frac{\partial C(\theta)^+}{\partial \theta_i} > \frac{\partial C(\theta)^-}{\partial \theta_i}$ hence θ_i will decrease and vice versa if the gradient is negative. Thus, there is a one-to-one relation between fixed points of the multiplicative update rule and stationary points under gradient descend. One attractive property of multiplicative updates is that, since θ_i, $\frac{\partial C(\theta)^+}{\partial \theta_i}$ and $\frac{\partial C(\theta)^-}{\partial \theta_i}$ all are non-negative, non-negativity is naturally enforced as each update remains in the positive orthant. α is a step size parameter that potentially can be tuned to assist convergence. When $\alpha \to 0$ only very small steps in the negative gradient direction are taken.

Using multiplicative updates Lee and Seung [2000] devised two algorithms for NMF. One based on least squares minimization (LS) corresponding to the approximation error being homoscedatic gaussian noise the other based on Kullback-Leibler divergence (KL) corresponding to Poisson noise. They further proved that these updates given at the top of Table 1 monotonically decrease the cost function C for $\alpha = 1$.

Although the estimation of $\mathbf{W}$ or $\mathbf{H}$ for fixed $\mathbf{H}$ or $\mathbf{W}$, respectively, is a convex problem, the combined estimation alternatingly solving for $\mathbf{W}$ and $\mathbf{H}$ is not guaranteed to find the global minima. Furthermore, a NMF decomposition is in general not unique [Donoho

and Stodden, 2003]: If the data does not adequately span the positive orthant a rotation of the solution is possible violating uniqueness. Consequently, constraints in the form of sparseness has proven useful such that the ambiguity is resolved taking the solution being the sparsest by some measure of sparseness [Hoyer, 2002, 2004; Eggert and Körner, 2004]. Eggert and Körner [2004] derived an efficient algorithm for Sparse NMF based on multiplicative updates by penalizing values in $\mathbf{H}$ by a function $C_{sparse}(\mathbf{H})$ while keeping $\mathbf{W}$ normalized such that the sparsity is not achieved simply by letting $\mathbf{H}$ go to zero while $\mathbf{W}$ goes to infinity. Making the reconstruction invariant to this normalization, i.e., $\widetilde{\mathbf{R}} = \widetilde{\mathbf{W}}\mathbf{H}$ where $\widetilde{\mathbf{W}}_{i,d} = \frac{\mathbf{W}_{i,d}}{\sqrt{\Sigma_i \mathbf{W}_{i,d}^2}} = \frac{\mathbf{W}_{i,d}}{\|\mathbf{W}_d\|_F}$, they found multiplicative updates for the LS-algorithm which can be extended to the KL-algorithm, see Table 1. In the following analysis we will use $C_{sparse}(\mathbf{H}) = \|\mathbf{H}\|_1$, i.e., an L_1-norm penalty. One attractive property of the L_1-norm is that it can function as a proxy for the L_0 norm, i.e., can minimize the number of non-zero elements while is does not change the convexity of the cost-function when estimating $\mathbf{H}$ for fixed $\mathbf{W}$ [Donoho, 2006]. Notice, $\frac{\partial C_{sparse}(\mathbf{H})}{\partial \mathbf{H}} = \mathbf{1}$.

Consider the non-negative Tucker model, i.e $\mathcal{X}$, $\mathcal{G}$ and $\mathbf{A}^{(n)}$ are all non-negative. By 'matrizicing' $\mathcal{X}^{I_1 \times I_2 \times ... \times I_N}$ into a matrix, i.e., $\mathbf{X}_{(n)}^{I_n \times I_1 ... I_{n-1} I_{n+1} ... I_N}$ the Tucker model can be expressed in matrix notation as [Lathauwer et al., 2000]

$$\mathbf{X}_{(n)} \approx \mathbf{R}_{(n)} = \mathbf{A}^{(n)} \mathbf{G}_{(n)} (\mathbf{A}^{(N)} \otimes ... \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes ... \otimes \mathbf{A}^{(1)}) = \mathbf{A}^{(n)} \mathbf{Z}_{(n)},$$

where $\mathbf{Z}_{(n)} = \mathbf{G}_{(n)} (\mathbf{A}^{(N)} \otimes ... \otimes \mathbf{A}^{(n+1)} \otimes \mathbf{A}^{(n-1)} \otimes ... \otimes \mathbf{A}^{(1)})^T$. As a result, the updates of each of the factors $\mathbf{A}^{(n)}$ follow straightforward from the regular NMF updates by exchanging $\mathbf{W}$ with $\mathbf{A}^{(n)}$ and $\mathbf{H}$ with $\mathbf{Z}_{(n)}$ in the $\mathbf{W}$ update.

By lexicographical indexing of the elements in $\mathcal{X}$ and $\mathcal{G}$, i.e., $vec(\mathcal{X})$ and $vec(\mathcal{G})$ the problem of finding the core $\mathcal{G}$ can be formulated in the framework of conventional factor analysis [Kolda, 2006]:

$$vec(\mathcal{X}) \approx vec(\mathcal{R}) = \mathbf{A} vec(\mathcal{G}),$$

where $\mathbf{A} = \mathbf{A}^{(1)} \otimes \mathbf{A}^{(2)} \otimes ... \otimes \mathbf{A}^{(N)}$. Consequently, the update of $\mathcal{G}$ follows by the regular NMF updates exchanging $\mathbf{W}$ with $\mathbf{A}$ and $\mathbf{H}$ with $vec(\mathcal{G})$ in the $\mathbf{H}$ update. Finally, this update can be expressed

$$C_{LS}(\mathbf{V},\mathbf{R}) = \frac{1}{2}\sum_{ij}(\mathbf{V}_{i,j}-\mathbf{R}_{i,j})^2$$

$$\mathbf{W} \leftarrow \mathbf{W} \bullet \left(\frac{\mathbf{V}\mathbf{H}^{\mathrm{T}}}{\mathbf{R}\mathbf{H}^{\mathrm{T}}}\right)^{.\alpha}, \quad \mathbf{H} \leftarrow \mathbf{H} \bullet \left(\frac{\mathbf{W}^{\mathrm{T}}\mathbf{V}}{\mathbf{W}^{\mathrm{T}}\mathbf{R}}\right)^{.\alpha}$$

$$C_{KL}(\mathbf{V},\mathbf{R}) = \sum_{ij}\mathbf{V}_{i,j}log\frac{\mathbf{V}_{i,j}}{\mathbf{R}_{i,j}} - \mathbf{V} + \mathbf{R}_{i,j}$$

$$\mathbf{W} \leftarrow \mathbf{W} \bullet \left(\frac{\frac{\mathbf{V}}{\mathbf{R}}\mathbf{H}^{\mathrm{T}}}{\mathbf{E}\mathbf{H}^{\mathrm{T}}}\right)^{.\alpha}, \quad \mathbf{H} \leftarrow \mathbf{H} \bullet \left(\frac{\mathbf{W}^{\mathrm{T}}\frac{\mathbf{V}}{\mathbf{R}}}{\mathbf{W}^{\mathrm{T}}\mathbf{E}}\right)^{.\alpha}$$

$$C_{SparseLS} = C_{LS}(\mathbf{V},\widetilde{\mathbf{R}}) + \beta C_{sparse}(\mathbf{H})$$

$$\mathbf{W} \leftarrow \widetilde{\mathbf{W}} \bullet \left(\frac{\mathbf{V}\mathbf{H}^{\mathrm{T}} + \widetilde{\mathbf{W}}diag(\mathbf{1}\cdot\widetilde{\mathbf{R}}\mathbf{H}^{\mathrm{T}} \bullet \widetilde{\mathbf{W}})}{\widetilde{\mathbf{R}}\mathbf{H}^{\mathrm{T}} + \widetilde{\mathbf{W}}diag(\mathbf{1}\cdot\widetilde{\mathbf{V}}\mathbf{H}^{\mathrm{T}} \bullet \widetilde{\mathbf{W}})}\right)^{.\alpha}$$

$$\mathbf{H} \leftarrow \mathbf{H} \bullet \left(\frac{\widetilde{\mathbf{W}}^{\mathrm{T}}\mathbf{V}}{\widetilde{\mathbf{W}}^{\mathrm{T}}\widetilde{\mathbf{R}} + \beta\frac{\partial C_{sparse}(\mathbf{H})}{\partial \mathbf{H}}}\right)$$

$$C_{SparseKL} = C_{KL}(\mathbf{V},\widetilde{\mathbf{R}}) + \beta C_{sparse}(\mathbf{H})$$

$$\mathbf{W} \leftarrow \widetilde{\mathbf{W}} \bullet \left(\frac{\frac{\mathbf{V}}{\widetilde{\mathbf{R}}}\mathbf{H}^{\mathrm{T}} + \widetilde{\mathbf{W}}diag(\mathbf{1}\cdot\mathbf{E}\mathbf{H}^{\mathrm{T}} \bullet \widetilde{\mathbf{W}})}{\mathbf{E}\mathbf{H}^{\mathrm{T}} + \widetilde{\mathbf{W}}diag(\mathbf{1}\cdot\frac{\mathbf{V}}{\widetilde{\mathbf{R}}}\mathbf{H}^{\mathrm{T}} \bullet \widetilde{\mathbf{W}})}\right)^{.\alpha}$$

$$\mathbf{H} \leftarrow \mathbf{H} \bullet \left(\frac{\widetilde{\mathbf{W}}^{\mathrm{T}}\frac{\mathbf{V}}{\widetilde{\mathbf{R}}}}{\widetilde{\mathbf{W}}^{\mathrm{T}}\mathbf{E} + \beta\frac{\partial C_{sparse}(\mathbf{H})}{\partial \mathbf{H}}}\right)$$

Table1. The NMF updates (top) and Sparse NMF updates (bottom) for both LS and KL minimization. $C_{sparse}(\mathbf{H})$ is the function used to penalize the elements in $\mathbf{H}$. While the updates for regular NMF as well as updates where sparseness is given by $C_{sparse}(\mathbf{H}) = \|\mathbf{H}\|_1$ have been proven to converge for $\alpha = 1$ the normalization invariant $\mathbf{W}$ update has not been proved convergent, however, in practise they are, and thus the update has been conjectured convergent for $\alpha = 1$ Eggert and Körner [2004].

in terms of n-mode multiplication since

$$\mathbf{A}^T vec(\mathcal{X}) = vec(\mathcal{X} \times_1 \mathbf{A}^{(1)^T} \times_2 \mathbf{A}^{(2)^T} \times_3 ... \times_N \mathbf{A}^{(N)^T}).$$

The algorithms for SN-TUCKER are summarized in Table 2. Here $diag(\mathbf{v})$ is a matrix having the vector $\mathbf{v}$ along the diagonal while $\mathbf{1}$ and $\mathcal{E}$ is a matrix and a tensor having ones in all indices. In the Sparse SN-TUCKER some modalities can be kept sparse while the rest are normalized. Consequently, each or some of the $\mathbf{A}^{(n)}$ or $\mathcal{G}$,

Algorithm outline for SN-TUCKER based on LS and KL minimization

1. Initialize all $\mathbf{A}^{(n)}$ and the core array $\mathcal{G}$ for instance by random.
2. For all n do

   LS-minimization:

   $\mathbf{R}_{(n)} = \mathbf{A}^{(n)}\mathbf{Z}_{(n)}$

   $\mathbf{A}^{(n)} \leftarrow \mathbf{A}^{(n)} \bullet \left(\frac{\mathbf{X}_{(n)}\mathbf{Z}_{(n)}^T}{\mathbf{R}_{(n)}\mathbf{Z}_{(n)}^T} \right)^{.\alpha}$

   KL-minimization:

   $\mathbf{R}_{(n)} = \mathbf{A}^{(n)}\mathbf{Z}_{(n)}$

   $\mathbf{A}^{(n)} \leftarrow \mathbf{A}^{(n)} \bullet \left(\frac{\left(\frac{\mathbf{X}_{(n)}}{\mathbf{R}_{(n)}}\right)\mathbf{Z}_{(n)}^T}{\mathbf{E}_{(n)}\mathbf{Z}_{(n)}^T} \right)^{.\alpha}$

3. $\mathcal{R} = \mathcal{G} \times_1 \mathbf{A}^{(1)} \times_2 \mathbf{A}^{(2)} \times_3 ... \times_N \mathbf{A}^{(N)}$

   LS-minimization:

   $\mathcal{B} = \mathcal{X} \times_1 \mathbf{A}^{(1)^T} \times_2 \mathbf{A}^{(2)^T} \times_3 ... \times_N \mathbf{A}^{(N)^T}$

   $\mathcal{C} = \mathcal{R} \times_1 \mathbf{A}^{(1)^T} \times_2 \mathbf{A}^{(2)^T} \times_3 ... \times_N \mathbf{A}^{(N)^T}$

   $\mathcal{G} \leftarrow \mathcal{G} \bullet \left(\frac{\mathcal{B}}{\mathcal{C}}\right)^{.\alpha}$

   KL-minimization:

   $\mathcal{D} = \frac{\mathcal{X}}{\mathcal{R}} \times_1 \mathbf{A}^{(1)^T} \times_2 \mathbf{A}^{(2)^T} \times_3 ... \times_N \mathbf{A}^{(N)^T}$

   $\mathcal{F} = \mathcal{E} \times_1 \mathbf{A}^{(1)^T} \times_2 \mathbf{A}^{(2)^T} \times_3 ... \times_N \mathbf{A}^{(N)^T}$

   $\mathcal{G} \leftarrow \mathcal{G} \bullet \left(\frac{\mathcal{D}}{\mathcal{F}}\right)^{.\alpha}$

4. Repeat from step 2 until some convergence criterion has been satisfied

Table2. Algorithms for SN-TUCKER based on LS and KL minimization. In step 1, we initialized the components by random but such that the amplitude of the randomly generated data covered all potential solutions by the initialization. In step 4, the convergence was defined as a relative change in cost function being less than 10^{-6} or when the algorithm had run for 2500 iterations

can be constrained to be sparse while re-normalizing the modalities that are not constrained. In conclusion, sparseness can be imposed in any combination of modalities including the core, while normalizing the remaining modalities. In Table 2 the updates are given when sparsifying or normalizing a given modality. Here $\|\mathcal{G}\|_F = \sqrt{\sum_{j_1 j_2 .... j_N} \mathcal{G}^2_{j_1,j_2,...,j_N}}$ that is $\|\cdot\|_F$ is the regular Frobenious norm for matrices and tensors, respectively, as defined in [Kolda, 2006] while $\|\mathcal{G}\|_1 = \sum_{j_1 j_2 .... j_N} \mathcal{G}_{j_1,j_2,...,j_N}$. When normalizing, each of the updated $\mathbf{A}^{(n)}$'s should be normalized after the update, i.e., $\widetilde{\mathbf{A}}_{i_n,d} = \frac{\mathbf{A}_{i_n,d}}{\|\mathbf{A}_d\|_F}$

while the core is normalized by $\widetilde{\mathcal{G}} = \frac{\mathcal{G}}{\|\mathcal{G}\|_F}$. Notice,

	Normalized	Sparse
LS	$\widetilde{\mathbf{A}}^{(n)} \bullet \left(\frac{\mathbf{X}_{(n)}\mathbf{Z}_{(n)}^T + \tilde{\mathbf{A}}^{(n)} diag(\mathbf{1} \cdot \tilde{\mathbf{R}}_{(n)}\mathbf{Z}_{(n)}^T \bullet \tilde{\mathbf{A}}^{(n)})}{\tilde{\mathbf{R}}_{(n)}\mathbf{Z}_{(n)}^T + \tilde{\mathbf{A}}^{(n)} diag(\mathbf{1} \cdot \mathbf{X}_{(n)}\mathbf{Z}^T \bullet \tilde{\mathbf{A}}^{(n)})} \right)^{\cdot\alpha}$	$\mathbf{A}^{(n)} \bullet \left(\frac{\mathbf{X}_{(n)}\mathbf{Z}_{(n)}^T}{\mathbf{R}_{(n)}\mathbf{Z}_{(n)}^T + \beta} \right)^{\cdot\alpha}$
KL	$\widetilde{\mathbf{A}}^{(n)} \bullet \left(\frac{\left(\frac{\mathbf{X}_{(n)}}{\tilde{\mathbf{R}}_{(n)}}\right)\mathbf{Z}_{(n)}^T + \tilde{\mathbf{A}}^{(n)} diag(\mathbf{1} \cdot \mathbf{E}\mathbf{Z}_{(n)} \bullet \tilde{\mathbf{A}}^{(n)})}{\mathbf{E}\mathbf{Z}_{(n)}^T + \tilde{\mathbf{A}}^{(n)} diag(\mathbf{1} \cdot \left(\frac{\mathbf{X}_{(n)}}{\mathbf{R}_{(n)}}\right)\mathbf{Z}_{(n)}^T \bullet \tilde{\mathbf{A}}^{(n)})} \right)^{\cdot\alpha}$	$\mathbf{A}^{(n)} \bullet \left(\frac{\left(\frac{\mathbf{X}_{(n)}}{\mathbf{R}_{(n)}}\right)\mathbf{Z}_{(n)}^T}{\mathbf{E}\mathbf{Z}_{(n)}^T + \beta} \right)^{\cdot\alpha}$
LS	$\widetilde{\mathcal{G}} \bullet \left(\frac{\mathcal{B} + \tilde{\mathcal{G}}\|\mathcal{C} \bullet \tilde{\mathcal{G}}\|_1}{\mathcal{C} + \tilde{\mathcal{G}}\|\mathcal{B} \bullet \tilde{\mathcal{G}}\|_1} \right)^{\cdot\alpha}$	$\mathcal{G} \bullet \left(\frac{\mathcal{B}}{\mathcal{C} + \beta} \right)^{\cdot\alpha}$
KL	$\widetilde{\mathcal{G}} \bullet \left(\frac{\mathcal{D} + \tilde{\mathcal{G}}\|\mathcal{E} \bullet \tilde{\mathcal{G}}\|_1}{\mathcal{F} + \tilde{\mathcal{G}}\|\mathcal{D} \bullet \tilde{\mathcal{G}}\|_1} \right)^{\cdot\alpha}$	$\mathcal{G} \bullet \left(\frac{\mathcal{D}}{\mathcal{F} + \beta} \right)^{\cdot\alpha}$

Table3. Updates when normalizing or imposing sparseness on the various modalities. Top row updates of $\mathbf{A}^{(n)}$, bottom row updates of the core $\mathcal{G}$

$$C_{LS}(\mathbf{X}_{(1)}, \mathbf{R}_{(1)}) = ... = C_{LS}(\mathbf{X}_{(N)}, \mathbf{R}_{(N)}) = C_{LS}(vec(\mathcal{X}), \mathbf{A}vec(\mathcal{G}))$$
$$C_{KL}(\mathbf{X}_{(1)}, \mathbf{R}_{(1)}) = ... = C_{KL}(\mathbf{X}_{(N)}, \mathbf{R}_{(N)}) = C_{KL}(vec(\mathcal{X}), \mathbf{A}vec(\mathcal{G})).$$

Each of the updates above minimize the same cost function. As a result, the convergence of the algorithms for SN-TUCKER without sparseness for $\alpha = 1$ follow straightforward from the convergence of the regular NMF updates given in [Lee and Seung, 2000] as the estimation takes the form of a sequence of regular factor analysis problems minimizing the same cost function. However, no such proof exists for updates for normalized variables [Eggert and Körner, 2004]. Although extensively tested we never experienced any lack of convergence of the updates above for the normalized variables for $\alpha = 1$. Had the updates diverged α could have been tuned to ensure convergence.

The proposed algorithms for SN-TUCKER are based on multiplicative updates and in summary have the following benefits

– The developed algorithms can reduce ambiguities of the non-negative decompositions by imposing sparseness in any combination of modalities.

– The non-negativity ensures that no cancellation is allowed and that the representations becomes part based [Lee and Seung, 1999]. This also often leads to clustering of the data [Ding et al., 2005].

– Overcomplete representations can be handled, for instance the core tensor can for some modalities be much larger than the original data tensor, while sparsity can help to avoid an overfit of the data.

– The updates can easily be adapted to consider only the non-zero elements in $\mathcal{X}$ reducing computational complexity for highly sparse data.

– The updates can enforce specific prior structure in the core or the loadings. For instance the core or some of the core elements can be fixed to implement known interactions in the model simply by omitting the updates for these specific elements.

– Missing data is often a problem, however missing values can be handled by introducing an indicator tensor $\mathcal{Q}$ of same size as $\mathcal{V}$ having ones where data is present and zeros where missing as demonstrated for regular NMF in [Zhang et al., 2006]. Replacing $\mathcal{X}$ by $\mathcal{Q} \bullet \mathcal{X}_{(n)}$, $\mathcal{R}$ with $\mathcal{Q} \bullet \mathcal{R}$ and $\mathcal{E}$ with $\mathcal{Q}$ in the updates above the influence of missing values are completely removed in the model estimation.

– Each iteration of the SN-TUCKER is $\mathcal{O}(I_1 I_2 \cdot ... \cdot I_N J_1 J_2 \cdot ... \cdot J_N)$, i.e., grows linearly with the product of the size of $\mathcal{X}$ and $\mathcal{G}$ making the cost per iteration relatively limited compared to existing algorithms for non-negative TUCKER decomposition. Alternative algorithms, e.g., require an iterative check of the violation of non-negativity [Bro and Andersson, 2000; Bro and Jong, 1997].

A drawback compared to the algorithm for non-negative constrained optimization such as [Bro and Jong, 1997] is that convergence can be slow, especially for small values of the regularization parameters β. Although the estimation of each variable in turn is a convex optimization problem, alternatingly solving for the components of the

various modalities is a non-convex problem. Thus, just as for regular NMF the SN-TUCKER is prone to local minima. To speed up the convergence, we have used overrelaxed bound optimization as proposed for regular NMF in Salakhutdinov et al. [2003].

Finally, we note that if we force the core to be the identity tensor the algorithm reduces to the algorithm for non-negative PARAFAC also named Positive Tensor Factorization (PTF) proposed in [Welling and Weber, 2001].

3 Results and Discussion

In the following Standard Tucker will denote the algorithm for Tucker estimation provided by the N-way toolbox Bro and Andersson [2000] while HOSVD corresponds to the Tucker algorithm described in Lathauwer et al. [2000]. Furthermore, convergence will be defined here as a relative change in cost function being less than 10^{-6} or when the algorithm has run for 2500 iterations.

The algorithms were first tested on a synthetic data set consisting of 5 images of logical operators mixed through two modalities. The data was generated such that a perfect non-negative decomposition was ambiguously defined. The result of the decomposition of the synthetic data can be seen in Figure 1. While the SN-TUCKER KL and LS algorithm near perfectly identifies all components the corresponding non-negative PARAFAC decomposition, with its diagonal restriction on the core, fails in identifying the components. For the PARAFAC model the true interactions between the components of the various modalities can not be accounted for. The Standard Tucker algorithm provided by the N-way toolbox also failed in estimating the correct components as non-negativity of the core in the current implementation of the toolbox was not implemented. Thus, if the core is not constrained although the interactions (core-elements) are non-negative the decomposition results in an erroneous decomposition of the data. Namely, a pattern results with significant cancellation effects in the core that account for the data in a random way. Thus, even though the correct model has both non-negative loadings and interactions an unconstrained core will resort to cancellation effect in order to account for the data which hampers the interpretability of the model.

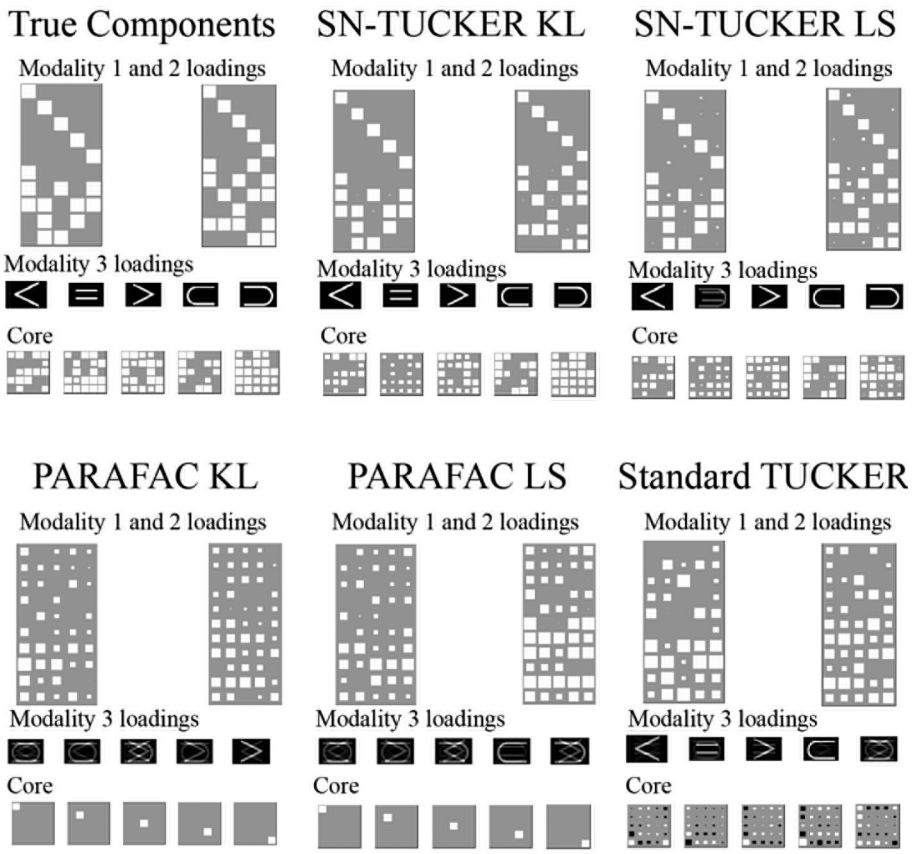

Figure1. Examples of results obtained when analyzing a synthetic data set generated from a Tucker 5-5-5 model. **Top left panel:** The true components generating the synthetic data. **Top middle panel:** Components obtained by the SN-TUCKER algorithm based on KL. **Top right panel:** Components obtained by the SN-TUCKER algorithm based on LS. **Bottom left panel:** Components obtained by the corresponding non-negative PARAFAC model based on KL. **Bottom middle panel:** Components obtained by the corresponding non-negative PARAFAC model based on LS. **Bottom right panel**: Components obtained by the Standard Tucker algorithm provided by the N-way toolbox (which is based on least squares minimization) allowing for the loadings to be constrained non-negative but keeping the core unconstrained. All decompositions except the PARAFAC decomposition accounts for more than 99.99% of the variance.

The algorithms were next tested on a data set containing the inter trial phase coherence (ITPC) obtained from wavelet transformed electroencephalographic (EEG) data. This data set has previously been analyzed using non-negative PARAFAC and a detailed description of the data set can be found in [Mørup et al., 2006]. Briefly stated it consist of 14 subject recorded during a proprioceptive stimuli consisting of a weight change of left hand during odd trials and right hand during even trials giving a total of $14 \cdot 2 = 28$ trials. Consequently, the data has the following form $\mathcal{X}_{ITPCvalue}^{Channel \times Time-Frequency \times Trials}$. The results of a Tucker 3-3-3 model can be seen in Figure 2 while an evaluation of the uniqueness of the decompositions is given in Table 4. Clearly, the SN-TUCKER model approaches the non-negative PARAFAC model as sparseness is imposed on the Core, see Figure 2. While the SN-TUCKER accounts for 49.3 % of the variance, the sparse SN-TUCKER accounts for 49.11 % of the variance whereas the non-negative PARAFAC model accounts for 48.9 % of the variance. Finally, the HOSVD accounts for 58.9 % of the variance while the two Standard Tucker decompositions both accounts for around 60 % of the variance. The decompositions constrained to be fully non-negative are easier to interpret compared to the HOSVD and decompositions based on Standard Tucker. The sparse SN-TUCKER and the PARAFAC decompositions are very similar both indicating a right sided and left sided activity in the first two components primarily during odd and even trials, respectively, corresponding to an activity contralateral to the stimulus side. The left and right sided activity represents information processing in the somatosensory and motor cortex situated in the parietal region of the brain contralateral to the stimulus side such that left hand is represented in the right hemisphere and vice versa for the right hand, see also [Mørup et al., 2006] for additional interpretation.

Since sparseness imposed on the core resulted in a decomposition resembling the corresponding PARAFAC decomposition we conclude that the PARAFAC rather than the full Tucker model can be considered a reasonable model to the data. Consequently, the Tucker model with sparsity imposed on the core can help to decide whether a PARAFAC or a Tucker model is the most appropriate model for a data set. Although, the decompositions obtained by the HOSVD and the standard Tucker procedure in the N-way toolbox accounts

for more variance since cancellation of factors are allowed, the decompositions are again harder to interpret. While the last factor in the *trial* modality clearly differentiates between left and right side stimulation and the second and third scalp components differentiates between frontal parietal and left right activity the interpretation of the interactions between these components are difficult to resolve from the complex pattern of interaction given by the cores. Consequently, although the SN-TUCKER model accounts for slightly less of the variance it is from an interpretation point of view more attractive. The SN-TUCKER is given for the LS minimization since this is the cost function the HOSVD and the Standard Tucker are based on.

β	0	1	10	100
LS	**Channel** : $F1$: 0.7416±0.2990 (0.3743±0.1352) $F2$: 0.8453±0.1032 (0.3328±0.0897) $F3$: 0.8401±0.0945 (0.3976±0.0814) **Time − Frequency** : $F1$: 0.8906±0.1937 (0.3175±0.0867) $F2$: 0.9317±0.0716 (0.3077±0.0674) $F3$: 0.9313±0.0729 (0.3126±0.0851) **Trials** : $F1$: 0.9268±0.0910 (0.4050±0.1131) $F2$: 0.9538±0.0480 (0.4055±0.1215) $F3$: 0.8661±0.1609 (0.4835±0.0965) **Core** : 0.7420±0.1048 (0.2853±0.1776) **Explained variance** : 0.4912±0.0027	**Channel** : $F1$: 0.9464±0.0471 (0.3427±0.0949) $F2$: 0.9492±0.0541 (0.3932±0.1072) $F3$: 0.9595±0.0381 (0.3660±0.1116) **Time − Frequency** : $F1$: 0.9753±0.0212 (0.3111±0.0378) $F2$: 0.9258±0.1254 (0.3108±0.0378) $F3$: 0.9368±0.1312 (0.3277±0.0484) **Trials** : $F1$: 0.9657±0.0222 (0.3465±0.1702) $F2$: 0.9585±0.1485 (0.4852±0.0674) $F3$: 0.9664±0.1161 (0.4620±0.0674) **Core** : 0.9139±0.0383 (0.2793±0.1244) **Explained variance** : 0.4909±0.0017	**Channel** : $F1$: 1.000±0.000 (0.3813±0.1400) $F2$: 1.000±0.000 (0.3636±0.1631) $F3$: 1.000±0.000 (0.3417±0.1072) **Time − Frequency** : $F1$: 1.000±0.000 (0.2812±0.0380) $F2$: 1.000±0.000 (0.3259±0.0661) $F3$: 1.000±0.000 (0.3329±0.0555) **Trials** : $F1$: 1.000±0.000 (0.4268±0.1402) $F2$: 1.000±0.000 (0.3897±0.1815) $F3$: 1.000±0.000 (0.3947±0.1375) **Core** : 0.6963±0.3535 (0.3473±0.1470) **Explained variance** : 0.3695±0.0000	**Channel** : $F1$: 1.000±0.000 (0.3428±0.1195) $F2$: 1.000±3.4700.000 (0.3657±0.1406) $F3$: 1.000±3.3870.000 (0.3914±0.1305) **Time − Frequency** : $F1$: 1.000±0.000 (0.3327±0.0398) $F2$: 1.000±0.000 (0.3288±0.0417) $F3$: 1.000±0.000 (0.2935±0.0210) **Trials** : $F1$: 1.000±0.000 (0.3681±0.0972) $F2$: 1.000±0.000 (0.4116±0.1434) $F3$: 1.000±0.000 (0.4507±0.1347) **Core** : 0.3561±0.1493 (0.3094±0.1141) **Explained variance** : −0.2600±0.0000

Table4. Mean correlation between the factors of 10 runs (stopped after 250 iterations) with sparseness imposed on the core array ranging from 0 to 100 here given for LS (range of data $[0; 0.4]$). In parenthesis are the correlations obtained by random (estimated by permutating the indices of the factors and calculating their correlation). Clearly imposing sparseness improves uniqueness (correlation between each decomposition) however if the sparseness imposed on the core is too strong all factors becomes identical only capturing the mean activity while the core is arbitrary due to the identical factors). The KL algorithm gave similar results.

From Table 4 we learn that each unconstrained SN-TUCKER decomposition is only inter-run correlated by about 70-90%. However, when imposing sparseness on the core a more unique decomposition was indeed achieved hence a correlation well above 90% between the components of the factors and core of the 10 decompositions while only slightly affecting the explained variance. However, by further increasing sparseness on the Core a new biased type of solution emerged in which a mean activity is represented in all the components. Consequently, the factors were all perfectly correlated to each other while the core could be arbitrarily chosen as long as the sum of the core elements remained the same leading to a high variant core and a useless decomposition.

Finally, the algorithms were tested on a data set of $\mathcal{X}_{Strength}^{Spectra \times Time \times Batch}$ obtained from a flow injection analysis (FIA) system, see [Nørgaard and Ridder, 1994; Smilde et al., 1999]. The data set has been analyzed through various supervised models using among other the prior knowledge of the concentration in each batch [Nørgaard and Ridder, 1994; Smilde et al., 1999]. However, here we employ a sparse SN-TUCKER to see if this algorithm can capture the underlying structure in the data unsupervised. Sparseness was imposed on both the core and batch modality ($\beta = 0.5$, range of data [0;0.637]). The results of the sparse Tucker 6-6-6 decomposition are given in Figure 3

From the analysis of the FIA data a highly consistent decomposition resulted when imposing sparseness on the core and batch modality, see Table 5. Here, the model captured the known true concentrations in the batch quite well while forming a sparse core also improved the interpretability of the components since less interactions were included, see Figure 3. Consequently, imposing sparseness can *turn off* excess factors, hence, assist model selection also capturing the true loadings as presently demonstrated by the decompositions ability to well estimate the known mixing profiles of the batches. Neither the decompositions without sparsity nor the Tucker procedure given in N-way toolbox allowing for negative core elements were as consistent nor were they able to capture well the true mixing. Furthermore, the corresponding 6 component non-negative PARAFAC decomposition was not able to identify the correct mixing as the model was inadequate for the data. Instead it seems that

	SN-TUCKER ($\beta = 0$)	SN-TUCKER ($\beta = 0.5$)	Standard TUCKER	PARAFAC
mean correlation (core and loadings)	0.8986 ± 0.0722 (0.3672 ± 0.1617)	0.9847 ± 0.0396 (0.4008 ± 0.1736)	0.9111 ± 0.0409 (0.3735 ± 0.1520)	0.9882 ± 0.0336 (0.4087 ± 0.1672)
mean correlation (est. and true mixing)	0.7588 ± 0.1460 (0.2984 ± 0.1979)	0.9550 ± 0.0648 (0.3258 ± 0.1863)	0.5478 ± 0.0870 (0.2387 ± 0.1963)	0.9391 ± 0.1032 (0.2648 ± 0.1814)
explained variance	$0.9995 \pm 1e^{-5}$	0.9972 ± 0.0007	$0.9997 \pm 2e^{-7}$	$0.9989 \pm 4e^{-4}$

Table5. Mean correlation of 10 decompositions of the FIA dataset for SN-TUCKER with and without sparseness as well as the Standard Tucker method and non-negative PARAFAC decomposition. In parenthesis are the correlations obtained by random (estimated by permutating the indices of the factors and calculating their correlation). Clearly, imposing sparseness improves component identification and reduce decomposition ambiguity while not hampering the models ability to account for the data. Correlation between estimated and true mixing is taken as the mean of the maximum correlation between each estimated component and the true components.

component 1 of the mixing matrix of the SN-TUCKER somewhat has been split into component 2 and 4, component 2 into 5 and 6 and component 3 into component 1 and 3 of the PARAFAC decomposition. Thus, the PARAFAC model is due to the restricted core forced to split the components of one mode that are shared by several components in another mode into duplicates of the same components. That the mixing components are duplicated in the PARAFAC decomposition can also be seen from the relative high correlation of the PARAFAC model to the true mixing as given in table 5. Thus, the SN-TUCKER model yield a more compact representation than the corresponding PARAFAC decomposition while imposing sparseness enables to capture the true structure in the data in a completely unsupervised manner, rather than resorting to supervised approaches as previously done [Nørgaard and Ridder, 1994; Smilde et al., 1999].

By forcing the structure of the core to be the identity tensor, the SN-TUCKER algorithm becomes an algorithm for the estimation of the PARAFAC model. Although, the PARAFAC model in general is unique under mild conditions [Kruskal, 1977], the PARAFAC model constrained to non-negativity is not in general unique [Lim and Golub, 2006]. Thus, imposing sparseness as presently proposed can also be used to alleviate the non-uniqueness of non-negative PARAFAC decompositions. The proposed SN-TUCKER has two

drawbacks. Estimating a good value of β is not obvious. Presently, we examined a few different values of β. Future work should investigate methods that more systematically estimate the β parameters such as approaches based on the L-curve [Hansen, 1992; Lawson and Hanson, 1974], generalized cross-validation [Golub et al., 1979] or Bayesian learning [Hansen et al., 2006]. Other approaches of tuning β have been to constrain the decompositions to give specific degree of sparseness [Hoyer, 2004; Heiler and Schnörr, 2006]. However, it is still not clear what degree of sparseness is desirable and as such the problem of choosing the regularization parameter β becomes the restated problem of choosing the correct sparsity degree. That is, there is a correspondence between sparsity degree as measured by $\frac{1}{\sqrt{I_n J_n}-1}(\sqrt{I_n J_n} - \frac{\|\mathbf{A}^{(n)}\|_1}{\|\mathbf{A}^{(n)}\|_2})$ and the value of β. Furthermore, while NMF and non-negative PARAFAC normally needs in the order of 100 iterations to get good solutions, to our experience the SN-TUCKER needs in the order of 1000 iterations, i.e., considerably more. The SN-TUCKER method was in general much slower than the HOSVD which has a closed form solution solving N eigenvalue problems. The decomposition was also considerably slower than the Standard Tucker method provided by the N-way toolbox and the non-negative PARAFAC proposed in [Welling and Weber, 2001]. However, for both the HOSVD as well as Standard Tucker the core can be directly calculated from pseudo-inverses of the loading matrices, i.e., as

$$\mathcal{G} = \mathcal{X} \times_1 \mathbf{A}^{(1)^\dagger} \times_2 \mathbf{A}^{(2)^\dagger} \times_3 ... \times_N \mathbf{A}^{(N)^\dagger}. \tag{6}$$

While for the non-negative PARAFAC no core is estimated. Thus, we also compared the present SN-TUCKER algorithm to an iterative procedure for fully non-negative Tucker (including non-negative core), extending the Standard Tucker algorithm provided by the N-way toolbox to include non-negative core updates based on the active set algorithm given in [Bro and Jong, 1997]. This significantly slowed down the algorithm making it comparable in time-usage to the SN-TUCKER algorithms we have proposed here. As a result, the SN-TUCKER model is considerably slower than Standard Tucker and non-negative PARAFAC due to the core update. Thus, future work should investigate how the convergence rate can be improved

when a closed form solution for the core no longer exists due to the non-negativity constraints.

4 Conclusion

We proposed two new sparse non-negative Tucker (SN-TUCKER) algorithms. Evidence was presented that SN-TUCKER yields a parts based representation as have been seen in NMF for 2-way data. Hence, a 'simpler', more interpretable decomposition than the decompositions obtained by current Tucker algorithms such as the HOSVD and the Standard Tucker algorithm provided by the N-way toolbox. Furthermore, imposing constraints of sparseness helped reduce ambiguities in the decomposition and turned off excess components, hence helped model selection and component identification. The analysis of the wavelet transformed EEG-data demonstrated how sparseness reduced ambiguities and can further be used to identify the adequacy of the PARAFAC model over the Tucker model. Whereas, the SN-TUCKER analysis of the FIA data demonstrated how sparseness not only improve uniqueness of the decompositions but is also able to turn of excess components such that the true loadings could be identified unsupervised and a more compact representation given than the representation obtained from the corresponding PARAFAC model. The algorithms presented can be downloaded from [Mørup, 2007].

References

Andersson, C. A. and Bro, R. (1998). Improving the speed of multiway algorithms: Part i. tucker3. *Chemometrics and Intelligent Laboratory Systems*, 42:93–103.

Bro, R. and Andersson, C. A. (2000). The n-way toolbox for matlab. *Chemometrics and Intelligent Laboratory Systems*, 52:1–4.

Bro, R. B. and Jong, S. D. (1997). A fast non-negativity-constrained least squares algorithm. *Journal of Chemometrics*, 11(5):393–401.

Carroll, J. D. and Chang, J. J. (1970). Analysis of individual differences in multidimensional scaling via an N-way generalization of "Eckart-Young" decomposition. *Psychometrika*, 35:283–319.

Cichocki, A., Zdunek, R., and Amari, S. (2006). Csiszar's divergences for non-negative matrix factorization: Family of new algorithms. *6th International Conference on Independent Component Analysis and Blind Signal Separation*, pages 32–39.

Cichocki, A., Zdunek, R., Choi, S., Plemmons, R., and Amari, S.-i. (2007). Nonnegative tensor factorization using alpha and beta divergencies. *ICASSP*.

Dhillon, I. S. and Sra, S. (2005). Generalized nonnegative matrix approximations with bregman divergences. *NIPS*, pages 283–290.

Ding, C., He, X., and Simon, H. D. (2005). On the equivalence of nonnegative matrix factorization and spectral clustering. Proc. SIAM Int'l Conf. Data Mining (SDM'05), pages 606–610.

Donoho, D. (2006). For most large underdetermined systems of linear equations the minimal l^1-norm solution is also the sparsest solution. *Communications on Pure and Applied Mathematics*, 59(6):797–829.

Donoho, D. and Stodden, V. (2003). When does non-negative matrix factorization give a correct decomposition into parts? *NIPS*.

Eggert, J. and Körner, E. (2004). Sparse coding and nmf. In *Neural Networks*, volume 4, pages 2529–2533.

FitzGerald, D., Cranitch, M., and Coyle, E. (2005). Non-negative tensor factorisation for sound source separation. In *proceedings of Irish Signals and Systems Conference*, pages 8–12.

Golub, G., Heath, M., and Wahba, G. (1979). Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, 21(2):215–223.

Gurden, S. P., Westerhuis, J. A., Bijlsma, S., and Smilde, A. K. (2001). Modelling of spectroscopic batch process data using grey models to incorporate external information. *Journal of Chemometrics*, 15:101–121.

Hansen, L. K., Madsen, K. H., and Lehn-Schiøler, T. (2006). Adaptive regularization of noisy linear inverse problems. In *Proceedings of Eusipco 2006*.

Hansen, P. C. (1992). Analysis of discrete ill-posed problems by means of the l-curve. *SIAM Review*, 34(4):561–580.

Harshman, R. A. (1970). Foundations of the PARAFAC procedure: Models and conditions for an "explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics*, 16:1–84.

Heiler M. and Schnörr, C. (2006). Controlling Sparseness in Non-Negative Tensor Factorization. *Lecture Notes in Computer Science*, 3951:56–67.

Hoyer, P. (2002). Non-negative sparse coding. *Neural Networks for Signal Processing, 2002. Proceedings of the 2002 12th IEEE Workshop on*, pages 557–565.

Hoyer, P. (2004). Non-negative matrix factorization with sparseness constraints. *Journal of Machine Learning Research* 5:1457–1469 .

Jia, K. and Gong, S. (2005). Multi-modal tensor face for simultaneous super-resolution and recognition. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1683–1690.

Kolda, T. G. (2006). Multilinear operators for higher-order decompositions. Technical Report SAND2006-2081, tr:sandreport.

Kruskal, J. (1977). Three-way arrays: rank and uniqueness of trilinear decompositions, with application to arithmetic complexity and statistics. *Linear Algebra Appl.*, 18:95–138.

Lathauwer, L. D., Moor, B. D., and Vandewalle, J. (2000). Multilinear singular value decomposition. *SIAM J. MATRIX ANAL. APPL.*, 21(4):1253Ű1278.

Lawson, C. and Hanson, R. (1974). *Solving Least Squares Problems*. Prentice-Hall.

Lee, D. and Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, 401(6755):788–91.

Lee, D., Seung, H., and Saul, L. (2002). Multiplicative updates for unsupervised and contrastive learning in vision. *Knowledge-Based Intelligent Information Engineering Systems and Allied Technologies. KES 2002*, 1:387–91.

Lee, D. D. and Seung, H. S. (2000). Algorithms for non-negative matrix factorization. In *NIPS*, pages 556–562.

Lim, L.-H. and Golub, G. (2006). Nonnegative decomposition and approximation of nonnegative matrices and tensors. *SCCM Technical Report, 06-01, forthcoming, 2006*.

Lin, C.-J. (2007). Projected gradient methods for non-negative matrix factorization. *To appear in Neural Computation*.

Mørup, M. (2007). Algorithms for SN-TUCKER. *www2.imm.dtu.dk/pubdb/views/edoc_ download.php/4718/zip/imm4718.zip*.

Mørup, M., Hansen, L. K., Parnas, J., and Arnfred, S. M. (2006). Decomposing the time-frequency representation of EEG using non-negative matrix and multi-way factorization. Technical report.

Murakami, T. and Kroonenberg, P. M. (2003). Three-mode models and individual differences in semantic differential data. *Multivariate Behavioral Research*, 38(2):247–283.

Nørgaard, L. and Ridder, C. (1994). Rank annihilation factor analysis applied to flow injection analysis with photodiode-array detection. *Chemometrics and Intelligent Laboratory Systems*, 23(1):107–114.

Olshausen, B. A. and Field, D. J. (2004). Sparse coding of sensorty inputs. *Current Opinion in Neurobiology*, 14:481–487.

Paatero, P. and Tapper, U. (1994). Positive matrix factorization: A non-negative factor model with optimal utilization of error estimates of data values. *Environmetrics*, 5(2):111–126.

Parry, Mitchell, R. and Essa, I. (2006). Estimating the spatial position of spectral components in audio. In *proceedings ICA2006*, pages 666– 673.

Salakhutdinov, R., Roweis, S., and Ghahramani, Z. (2003). On the convergence of bound optimization algorithms. In *Proceedings of the 19th Annual Conference on Uncertainty in Artificial Intelligence (UAI-03)*, pages 509–516.

Sidiropoulos, N. D. and Bro, R. (2000). On the uniqueness of multilinear decomposition of n-way arrays. *Journal of Chemometrics*, 14:229–239.

Smaragdis, P. and Brown, J. C. (2003). Non-negative matrix factorization for polyphonic music transcription. *IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, pages 177–180.

Smilde, A., Bro, R., and Geladi, P. (2004). *Multi-way Analysis: Applications in the Chemical Sciences*. Wiley.

Smilde, A. K. S., Tauller, R., Saurina, J., and Bro, R. (1999). Calibration methods for complex second-order data. *Analytica Chimica Acta*, 398:237–251.

Sun, J.-T., Zeng, H.-J., Liu, H., Lu, Y., and Chen, Z. (2005). Cubesvd: a novel approach to personalized web search. In *WWW '05: Proceedings of the 14th international conference on World Wide Web*, pages 382–390.

Tucker, L. R. (1966). Some mathematical notes on three-mode factor analysis. *Psychometrika*, 31:279–311.

Vasilescu, M. A. O. and Terzopoulos, D. (2002). Multilinear analysis of image ensembles: Tensorfaces. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 447–460.

Wang, H. and Ahuja, N. (2003). Facial expression decomposition. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, 2:958–965.

Welling, M. and Weber, M. (2001). Positive tensor factorization. *Pattern Recogn. Lett.*, 22(12):1255–1261.

Zhang, S., Wang, W., Ford, J., and Makedon, F. (2006). Learning from incomplete ratings using non-negative matrix factorization. *6th SIAM Conference on Data Mining (SDM)*, pages 548–552.

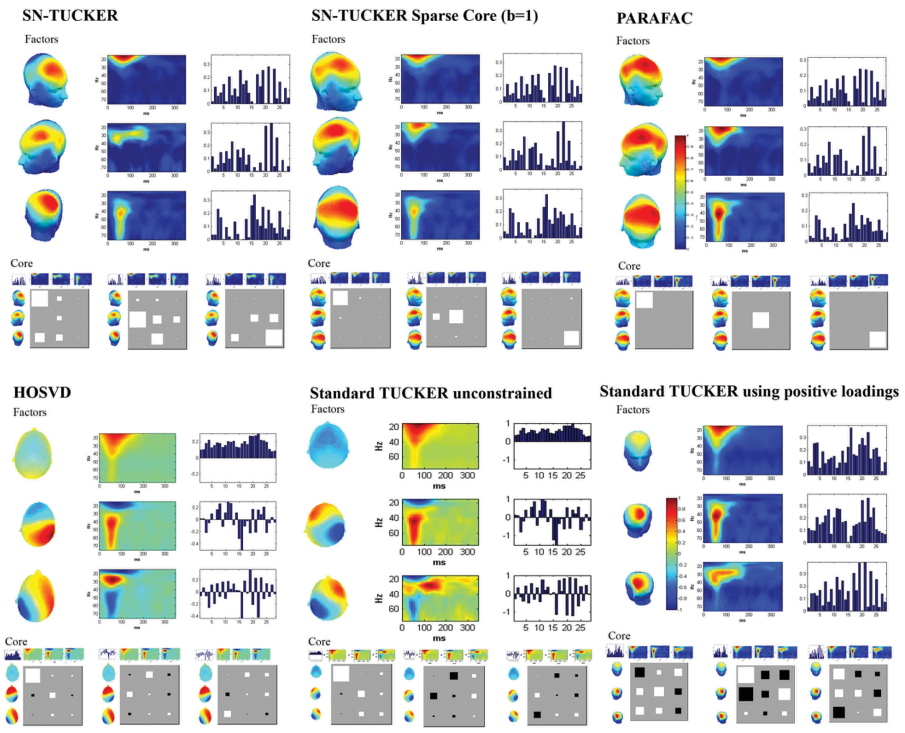

Figure2. Analysis of the ITPC data of EEG consisting of 14 subjects undergoing weight change of left hand during odd trials and right hand during even trials. **Top left panel:** Example of result obtained when analyzing the data using SN-TUCKER. **Top middle panel:** Result when imposing sparseness on the core ($\beta = 1$, range of data [0;0.4]). **Top right panel:** The results obtained from the PARAFAC model corresponding to a fixed Core having ones along the diagonal. **Bottom left panel:** The results obtained using HOSVD. **Bottom middle panel:** Results obtained using the Standard Tucker procedure provided by the N-way toolbox without constraints. **Bottom right panel:** Results obtained using Standard Tucker imposing non-negativity on all the loadings.

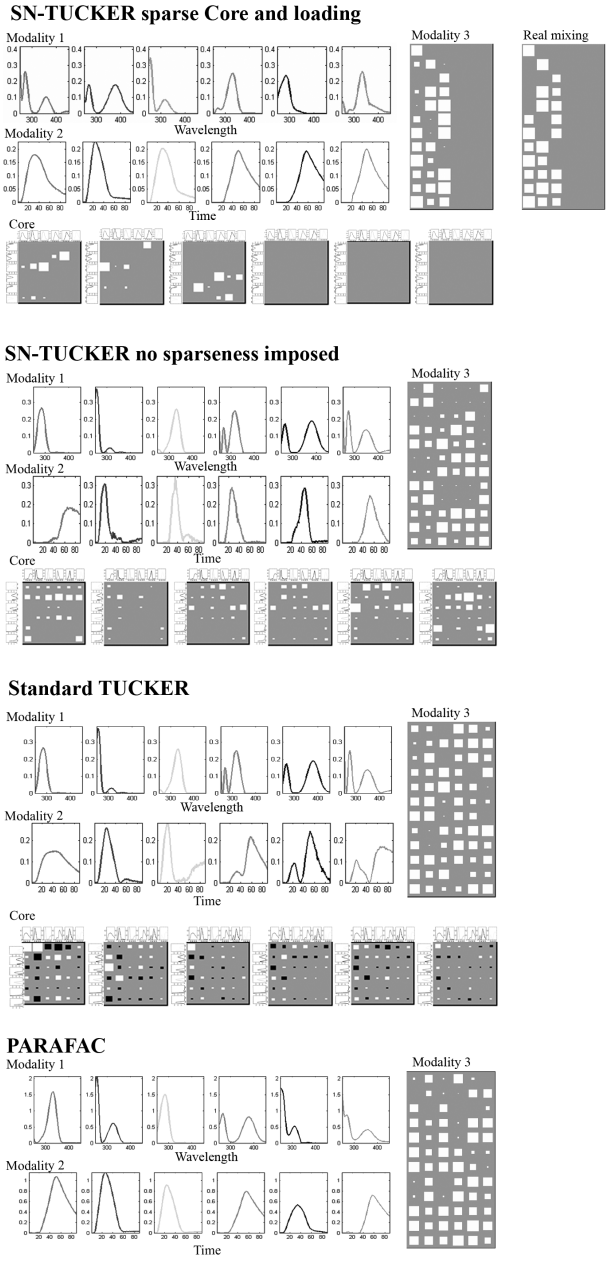

Figure3. The result obtained analyzing the FIA data by a Tucker 6-6-6 model. **Top panel:** SN-TUCKER based on LS with sparsity on the Core and mixing modality, ($\beta = 0.5$ range of data [0; 0.637]). **Upper middle panel:** Example of result obtained by a SN-TUCKER with no sparsity imposed. **Lower middle panel:** Example of decomposition obtained using the Standard Tucker procedure provided by the N-way toolbox imposing non-negativity on the loadings. The SN-TUCKER presently used LS minimization since this is the cost function the Standard Tucker also minimizes. **Bottom panel:** Result obtained from the corresponding 6 component non-negative PARAFAC decomposition.