Data Min Knowl Disc (2011) 22:419–466
DOI 10.1007/s10618-010-0196-4

Nonnegative tensor factorization as an alternative Csiszar–Tusnady procedure: algorithms, convergence, probabilistic interpretations and novel probabilistic tensor latent variable analysis algorithms

Stefanos Zafeiriou · Maria Petrou

Received: 1 May 2009 / Accepted: 8 July 2010 / Published online: 1 August 2010

Abstract In this paper we study Nonnegative Tensor Factorization (NTF) based on the Kullback–Leibler (KL) divergence as an alternative Csiszar–Tusnady procedure. We propose new update rules for the aforementioned divergence that are based on multiplicative update rules. The proposed algorithms are built on solid theoretical foundations that guarantee that the limit point of the iterative algorithm corresponds to a stationary solution of the optimization procedure. Moreover, we study the convergence properties of the optimization procedure and we present generalized pythagorean rules. Furthermore, we provide clear probabilistic interpretations of these algorithms. Finally, we discuss the connections between generalized Probabilistic Tensor Latent Variable Models (PTLVM) and NTF, proposing in that way algorithms for PTLVM for arbitrary multivariate probabilistic mass functions.

Keywords Nonnegative Matrix Factorization · Nonnegative Tensor Factorization · Kullback–Leibler divergence · Probabilistic Latent Semantic Analysis

1 Introduction

Nonnegative Matrix Factorization (NMF) has been proposed for the analysis of nonnegative data in Lee and Seung (1999, 2000). In Lee and Seung (1999), NMF was motivated as a technique for discovering the "parts" of objects. This was shown when

Responsible editors: Tao Li, Chris Ding, Fei Wang.

S. Zafeiriou (✉) · M. Petrou
Department of Electrical and Electronic Engineering, Imperial College London, Signal Processing and Communication Research Group, South Kensington Campus, London SW7 2AZ, UK
e-mail: s.zafeiriou@imperial.ac.uk

NMF was applied to the decomposition of facial images. It was demonstrated that the derived bases of the decomposition intuitively corresponded to "parts" of faces.

The problem of NMF is as follows: consider a nonnegative matrix $\mathbf{X} \in \Re_+^{F \times L}$ and a prespecified integer constant K, and to find a matrix $\mathbf{Z} \in \Re_+^{F \times K}$, which is the matrix that contains the bases of the decomposition in its columns (Lee and Seung 1999, 2000; Zafeiriou et al. 2006; Kotsia et al. 2007), and a matrix $\mathbf{H} \in \Re_+^{K \times L}$, which contains the weights of the decomposition, such that $\mathbf{X} \approx \mathbf{ZH}$. In order to find the approximation $\mathbf{X} \approx \mathbf{ZH}$ two optimization problems were defined in Lee and Seung (2000): the first optimization problem minimizes the Frobenius norm between $\mathbf{X}$ and the decomposition $\mathbf{ZH}$ and yields a least squares approximation, while the second one resorts to the minimization of the Kullback–Leibler (KL) divergence between $\mathbf{X}$ and the decomposition. Updating rules of the iterative schemes used to solve these problems were derived from auxiliary functions. The provided updating rules guaranteed the non-increasing behavior of the cost functions, but they did not guarantee that the limit point corresponded to a local minimum or to a stationary limit point (Gonzalez and Zhang 2005; Lin 2007a,b).

In Gonzalez and Zhang (2005) it was numerically demonstrated that the multiplicative update rules in Lee and Seung (1999, 2000) may fail to converge to a stationary point. Recently, the convergence properties of the NMF update rules were explored in Lin (2007a,b) and Finesso and Spreij (2006). More precisely, in Lin (2007a) update rules, which guarantee that the limit point of the Least Squares Error (LSE) optimization problem is a stationary point, were proposed. In Lin (2007b) an alternative solution for the LSE optimization problem was proposed that was based on the Armijo rule. In both Lin (2007a,b) LSE was used for measuring the error of the approximation.

Another very useful measure for the error of the nonnegative approximation with many applications is the KL divergence. The first algorithm with the KL divergence was proposed in Lee and Seung (2000) where both multiplicative and additive update rules were proposed. In Sra and Dhillon (2006), using the generalized Bregman distance formulation, an exponential family of update rules was proposed for the KL divergence. In Finesso and Spreij (2006), a convergence analysis for the multiplicative update rules of the KL divergence was provided and NMF was interpreted within a probabilistic framework. Moreover, in Finesso and Spreij (2006), it was proven that the proposed set of multiplicative update rules guarantee that the KL divergence converges to a stationary limit point. Finally, the relation between multiplicative update rules for NMF with KL divergence and Probabilistic Latent Semantic Analysis (PLSA) was explored in Gaussier and Goutte (2005). Recently, the relationship of NMF approaches with various Probabilistic Latent Variable Models (PLVM) was investigated in Shashanka et al. (2007, 2008) and Smaragdis and Raj (2007).

One disadvantage of NMF is that it cannot describe more than pairwise data relations. The application of NMF to modelling relations and furthermore to clustering was initiated in Lee and Seung (1999). In Ding et al. (2005) it was shown that there is a close relationship between spectral clustering and NMF. In Shashua et al. (2006) the problem of clustering data, given complex relations (beyond pairwise relations) between data points, was considered. The n-wise relations between the data points can be modelled by an n-order tensor, where each entry corresponds to an affinity mea-

sure (usually nonnegative) over an n-tuple of data points, which are more naturally described by a n-order tensor. That way, NTF models for clustering n-wise relations were motivated.

Another example of the advantage of NTF over NMF can be found when these methods are applied to feature extraction from images. In this case, in order to apply NMF, the object images should be vectorized in order to find their nonnegative decomposition. This vectorization leads to information loss, since the local structure of the images is lost. In order to remedy this drawback of NMF representation, the n-order and 3-order Nonnegative Tensor Factorization (NTF) schemes were proposed (Hazan et al. 2005; Friedlander and Hatz 2006; Shashua and Hazan 2005). In Shashua and Hazan (2005), an n-order tensor factorization method was proposed, based on the Frobenius norm. In Hazan et al. (2005), an image database was represented as a 3-order tensor, i.e. a 3D cube that has as slices the 2D images. Update rules for the factors (used in the decomposition), that guarantee a nonincreasing behavior of the KL divergence, were proposed. The 3D NTF decomposition was proven to be more suitable for part-based object representation than NMF (Hazan et al. 2005). Examples [like the decomposition of the Swimmer dataset (Donoho and Stodden 2004)] which demonstrate the superiority of the 3D NTF over the NMF can be found in Hazan et al. (2005). A recent overview of NTF algorithms can be found in Zafeiriou (2009b).

Recently, there has been an increasing interest in NTF algorithms (Kim and Choi 2007; Cichocki et al. 2007, 2008; Mørup et al. 2008; Kim et al. 2008; Zafeiriou 2009a,b). In Kim and Choi (2007), Mørup et al. (2008), and Kim et al. (2008) algorithms for arbitrary order Tucker (1966) factorizations, where data, core and mode matrices are non-negative, were proposed. Multiplicative update rules for all the factors were also proposed. In order to reduce further the ambiguities of the decomposition, updates that can impose sparseness in any combination of modalities were proposed in Mørup et al. (2008). Moreover, the notion of nonsmoothness (Pascual-Montano et al. 2006) for controlling the sparseness was extended for NTF algorithms in Kim and Choi (2007). Algorithms for 3D NTF using the PARAFAC2 (Kiers et al. 1999; Bro et al. 1999) decomposition were proposed in Cichocki et al. (2007, 2008). Finally, in Zafeiriou (2009a) the NMF method, that uses the generalized Bregman divergence in Sra and Dhillon (2006) and the Discriminant-NMF (DNMF) method in Zafeiriou et al. (2006), were generalized for arbitrary nonnegative tensor decomposition.

The tensorization of well-established vector-based algorithms is not an easy procedure and is a very active research field. For example, in De Lathauwer et al. (2000) Singular Value Decomposition (SVD) was extended to a multilinear SVD, in Lu et al. (2008) Principal Component Analysis (PCA) was extended to multilinear PCA, in Yan et al. (2007) and Tao et al. (2007) Fisher's Linear Discriminant Analysis (FLDA) was extended to various multilinear counterparts, Independent Component Analysis (ICA) was extended to multilinear ICA in Raj and Bovik (2008) and Canonical Correlation Analysis (CCA) to multilinear CCA in Kim and Cipolla (2008).

In this paper, we study NTF using the KL-divergence. The KL divergence is one of the most used distances for performing NMF and NTF and is of importance since it provides factorizations that share a lot of similarities with Expectation Maximization (EM) approaches. Moreover, such NMF and NTF approaches, in most cases,

lead to clear probabilistic interpretations. Algorithms for NTF based of KL divergence were recently proposed in Zafeiriou (2009a). In Zafeiriou (2009a) generalization of the NTF algorithms in Shashua and Hazan (2005) and of the NMF algorithms in Zafeiriou et al. (2006) and Sra and Dhillon (2006) were proposed. In Zafeiriou (2009a) no probabilistic interpretation for the NTF algorithm have been given and the properties of the algorithms have not been studied. In this paper, we interpret the NTF algorithm as an alternative Csiszar–Tusnady procedure, in a similar manner as Finesso and Spreij (2006), and we explore the convergence properties of the optimization procedure. This analysis leads to updating rules which guarantee that the limit point is stationary. Moreover, we investigate the relationship between Probabilistic Tensor Latent Variable Models (PTLVM) and NTF with the KL divergence and we propose novel algorithms for PTLVM. In Shashanka et al. (2008), Gaussier and Goutte (2005), and Ding et al. (2008), the relationship between Probabilistic Latent Variable Analysis (PLVA) models for bivariate distributions and NMF were explored. In Gaussier and Goutte (2005) a first attempt to connect NMF and PLSA was performed. In Ding et al. (2008) it was shown that NMF is similar but not exactly equivalent to NMF and a hybrid algorithm for performing PLSA using NMF was proposed. In Shashanka et al. (2008), it was also noted that the approach can be extended to arbitrary order PTLVM but no algorithm was proposed that could solve such problems. In this paper, we show the close connection between the algorithms in Finesso and Spreij (2006), Ding et al. (2008), Gaussier and Goutte (2005) and Shashanka et al. (2008) and we generalize them for arbitrary multivariate distributions and for arbitrary order tensors. Summarizing the novel contributions of this paper are:

- We propose novel nonnegative tensor factorization algorithms based on KL-divergence by interpreting the optimization problem as a alternative Csiszar–Tusnady procedure.
- We provide clear probabilistic interpretations for nonnegative tensor factorization, study the convergence and other theoretical properties of the algorithm.
- We propose algorithms for symmetric and asymmetric probabilistic tensor latent variable analysis.

All the algorithms are proposed using simple matrix operations.

The remainder of this paper is organized as follows. In Sect. 2, we briefly describe the problem of NMF with the KL divergence and comment on the relationship between NMF and PLVM. In Sect. 3, we briefly outline some elements of multilinear algebra and show how NMF algorithms can be extended to arbitrary order NTF. In Sect. 4, robust update rules of NTF with the KL divergence are proposed and their convergence theorems are presented. In Sect. 5 we present a probabilistic interpretation of the optimization problem and explore various properties of this approach. In Sect. 6, we show the equivalence between the proposed NTF algorithm and PTLVMs and propose algorithms for arbitrary order probability tensor decompositions. Experimental results which demonstrate the merits of the proposed approach are presented in Sect. 7. Finally, conclusions are drawn in Sect. 8. For completeness, extensive proofs of the various statements throughout the paper are given in the Appendices.

2 Nonnegative Matrix Factorization to NTF

In this section, we briefly describe how NMF is formulated. In the following, we consider a collection of L nonnegative vectors $\mathbf{x} \in \Re_+^F$.

2.1 Nonnegative Matrix Factorization

For two vectors $\mathbf{x} = [x_1, x_2, \ldots, x_F]^T$ and $\mathbf{q} = [q_1, q_2, \ldots, q_F]^T$, the KL divergence (or relative entropy) between them is defined as (Lee and Seung 2000):

$$KL(\mathbf{x}||\mathbf{q}) \triangleq \sum_{i=1}^{F} \left(x_i \ln \frac{x_i}{q_i} + q_i - x_i \right). \tag{1}$$

For convenience, let us define $\frac{0}{0} = 0$ and $0 \ln 0 = 0$ (Finesso and Spreij 2006). It can be shown, that, in general, the KL divergence is nonnegative and is equal to zero if and only if its two arguments are equal. The basic idea behind NMF is to approximate the object $\mathbf{x}$ by a linear combination of the elements of $\mathbf{h} \in \Re_+^K$ such that $\mathbf{x} \approx \mathbf{Zh}$, where $\mathbf{Z} \in \Re_+^{F \times K}$ is a nonnegative matrix, the columns of which sum up to 1. In order to measure the error of approximation $\mathbf{x} \approx \mathbf{Zh}$, the $KL(\mathbf{x}||\mathbf{Zh})$ divergence may be used (Lee and Seung 2000).

NMF, when applied to matrix $\mathbf{X} = [\mathbf{x}_1| \ldots |\mathbf{x}_j| \ldots |\mathbf{x}_L] \in \Re_+^{F \times L} = [x_{ij}]$, aims at finding two matrices $\mathbf{Z} \in \Re_+^{F \times M} = [z_{ik}]$ and $\mathbf{H} \in \Re_+^{M \times L} = [h_{k,j}]$ such that:

$$\mathbf{X} \approx \mathbf{ZH}. \tag{2}$$

Vector $\mathbf{x}_j$ after the NMF decomposition can be written as $\mathbf{x}_j \approx \mathbf{Zh}_j$, where $\mathbf{h}_j$ is the jth column of $\mathbf{H}$. Thus, the columns of matrix $\mathbf{Z}$ define a basis and $\mathbf{h}_j$ consists of the corresponding weights.

Decomposition (2) induces an approximation error which is the sum of all KL divergences for all $\mathbf{x}_j$, i.e.:

$$\begin{aligned} D_{\mathrm{KL}}(\mathbf{X}||\mathbf{ZH}) &= \sum_{j=1}^{L} KL(\mathbf{x}_j||\mathbf{Zh}_j) \\ &= \sum_{i=1}^{F} \sum_{j=1}^{L} \left(x_{ij} \ln \left(\frac{x_{ij}}{\sum_{k=1}^{K} z_{ik} h_{kj}} \right) + \sum_{k=1}^{K} z_{ik} h_{kj} - x_{ij} \right) \end{aligned} \tag{3}$$

as the measure of the cost for factoring $\mathbf{X}$ into $\mathbf{ZH}$ (Lee and Seung 2000).

Factorization (2) is the outcome of the following optimization problem:

$$\min_{\mathbf{Z},\mathbf{H}} D_{\mathrm{KL}}(\mathbf{X}||\mathbf{ZH}) \text{ subject to}$$
$$z_{ik} \geq 0, \; h_{kj} \geq 0, \; \forall k = 1, \ldots, M \text{ and } \forall i = 1, \ldots, F \quad \forall j = 1, \ldots, L.$$

NMF has non-negative constraints in both the elements of $\mathbf{Z}$ and $\mathbf{H}$; these nonnegativity constraints permit the combination of multiple basis components in order to represent the original nonnegative vectors using only additions between the different basis components. The following update rules guarantee a nonincreasing behavior of the KL divergence (3) (Lee and Seung 1999, 2000):

$$z_{ik}^{(t)} = z_{ik}^{(t-1)} \frac{\sum_{m=1}^{M} \left(h_{km}^{(t-1)} x_{im} \right) / [\mathbf{Z}^{(t-1)} \mathbf{H}^{(t-1)}]_{im}}{\sum_{m=1}^{M} h_{km}^{(t-1)}} \tag{4}$$

$$h_{kj}^{(t)} = h_{kj}^{(t-1)} \frac{\sum_{l=1}^{F} \left(z_{lk}^{(t)} x_{lj} \right) / [\mathbf{Z}^{(t)} \mathbf{H}^{(t-1)}]_{lj}}{\sum_{l=1}^{F} z_{lk}^{(t)}}. \tag{5}$$

2.2 Alternative multiplicative update rules of NMF using KL divergence and a probabilistic interpretation

In Finesso and Spreij (2006), an alternative algorithm for solving the optimization problem of minimizing (3) under nonnegativity constraints was proposed. That is, the problem was transformed in the factorization of a probabilities matrix $\mathbf{P}$ with $\mathbf{P}_{ij} \geq 0$ and $\sum_{i=1}^{F} \sum_{j=1}^{L} \mathbf{P}_{ij} = 1$. In the following for denoting the elements of matrix $\mathbf{P}$ we shall use either p_{ij} or $\mathbf{P}(i, j)$.

The problem was formulated as follows:

$$\min_{\mathbf{Q}_-, \mathbf{Q}_+ : \mathbf{Q}_+ \mathbf{e} = \mathbf{e}, \mathbf{e}^T \mathbf{Q}_- \mathbf{e} = 1} D_{\mathrm{KL}}(\mathbf{P} || \mathbf{Q}_- \mathbf{Q}_+) \tag{6}$$

for notation simplicity, we shall use only $\mathbf{e}$ for denoting an arbitrary dimensional vector of ones. For the above constrained optimization problem cost function (6) is simplified as:

$$D_{\mathrm{KL}}(\mathbf{P} || \mathbf{Q}_- \mathbf{Q}_+) = \sum_{i=1}^{F} \sum_{j=1}^{L} \left(\mathbf{P}(i, j) \ln \left(\frac{\mathbf{P}(i, j)}{\sum_{k=1}^{K} \mathbf{Q}_-(i, k) \mathbf{Q}_+(k, j)} \right) \right). \tag{7}$$

The update rules for solving the above optimization problem are:

$$\mathbf{Q}_-^{(t)}(i, k) = \mathbf{Q}_-^{(t-1)}(i, k) \sum_{m=1}^{M} \frac{\mathbf{Q}_+^{(t-1)}(k, m) \mathbf{P}(i, m)}{\left[\mathbf{Q}_-^{(t-1)} \mathbf{Q}_+^{(t-1)} \right]_{im}} \tag{8}$$

and

$$\mathbf{Q}_+^{(t)}(k, j) = \mathbf{Q}_+^{(t-1)}(k, j) \times \left(\sum_{i=1}^{F} \frac{\mathbf{Q}_-^{(t)}(i, k) \mathbf{P}(i, j)}{\left[\mathbf{Q}_-^{(t)} \mathbf{Q}_+^{(t-1)} \right]_{ij}} \right) \Bigg/ \left(\sum_{m=1}^{F} \sum_{l=1}^{M} \frac{\mathbf{Q}_-^{(t)}(m, k) \mathbf{Q}_+^{(t-1)}(k, l) \mathbf{P}(m, l)}{\left[\mathbf{Q}_-^{(t)} \mathbf{Q}_+^{(t-1)} \right]_{ml}} \right). \tag{9}$$

Update rule (9) may be equivalently written as

$$\mathbf{Q}_+^{(t)}(k,\,j) = \mathbf{Q}_+^{(t-1)}(k,\,j)\left(\sum_{i=1}^{F}\frac{\mathbf{Q}_-^{(t)}(i,k)\mathbf{P}(i,\,j)}{\left[\mathbf{Q}_-^{(t)}\mathbf{Q}_+^{(t-1)}\right]_{ij}}\right)\Big/\left(\sum_{l=1}^{N}\mathbf{Q}_-^{(t)}(l,k)\right). \quad (10)$$

Matrices $\mathbf{P}$, $\mathbf{Q}_-$ and $\mathbf{Q}_+$ are related with $\mathbf{X}$, $\mathbf{Z}$ and $\mathbf{H}$ through:

$$\mathbf{P} = \frac{1}{\mathbf{e}^T\mathbf{X}\mathbf{e}}\mathbf{X}, \qquad \mathbf{Q}_- = \frac{1}{\mathbf{e}^T\mathbf{Z}\mathbf{e}}\mathbf{Z}, \qquad \mathbf{Q}_+ = \mathbf{H}. \quad (11)$$

The above updating rules not-only guarantee the nonincreasigness of cost function $D_{\mathrm{KL}}(\mathbf{P}||\mathbf{Q}_-\mathbf{Q}_+)$ but also guarantee the stationarity of the limit point. By transforming the optimization problem (4) into the factorization of a probabilities matrix, not only helped in proving stationarity, but also revealed a lot of interesting properties and interpretations. The generalization of these properties to arbitrary order tensor factorizations will be explored in this paper.

Before proceeding we will briefly comment on an alternative factorization introducing another vector $\mathbf{a}$ which is also computed during the procedure. The optimization problem is:

$$\min_{\mathbf{Q}_-,\mathbf{Q}_+:\mathbf{Q}_+\mathbf{e}=\mathbf{e},\mathbf{Q}_-^T\mathbf{e}=\mathbf{e},\mathbf{a}^T\mathbf{e}=1} D_{\mathrm{KL}}(\mathbf{P}||\mathbf{Q}_-\mathbf{A}\mathbf{Q}_+) \quad (12)$$

where $\mathbf{A} = \mathbf{diag}(\mathbf{a})$ and $\mathbf{diag}(\mathbf{a})$ returns a diagonal matrix that has in its main diagonal the elements of vector $\mathbf{a}$. The corresponding update rules are given by:

$$\tilde{\mathbf{Q}}_-^{(t)}(i,k) = \mathbf{Q}_-^{(t-1)}(i,k)\sum_{m=1}^{M}\frac{\left[\mathbf{A}^{(t-1)}\mathbf{Q}_+^{(t-1)}\right]_{km}\mathbf{P}(i,m)}{\left[\mathbf{Q}_-^{(t-1)}\mathbf{A}^{(t-1)}\mathbf{Q}_+^{(t-1)}\right]_{im}} \quad (13)$$

$$\mathbf{Q}_-^{(t)}(i,k) = \frac{\tilde{\mathbf{Q}}_-^{(t)}(i,k)}{\sum_{m=1}^{F}\tilde{\mathbf{Q}}_-^{(t)}(m,k)},$$

$$\tilde{\mathbf{Q}}_+^{(t)}(k,\,j) = \mathbf{Q}_+^{(t-1)}(k,\,j)\left(\sum_{i=1}^{F}\frac{\left[\mathbf{Q}_-^{(t)}\mathbf{A}^{(t-1)}\right]_{ik}\mathbf{P}(i,\,j)}{\left[\mathbf{Q}_-^{(t)}\mathbf{A}^{(t-1)}\mathbf{Q}_+^{(t-1)}\right]_{ij}}\right)$$

$$\mathbf{Q}_+^{(t)}(k,\,j) = \frac{\tilde{\mathbf{Q}}_+^{(t)}(k,\,j)}{\sum_{m=1}^{L}\tilde{\mathbf{Q}}_+^{(t)}(m,k)}, \quad (14)$$

and

$$\mathbf{a}^{(t)}(k) = \mathbf{a}^{(t-1)}(k)\sum_{i=1,\,j=1}^{F,L}\frac{\mathbf{P}(i,\,j)\mathbf{Q}_+^{(t)}(k,\,j)\mathbf{Q}_-^{(t)}(k,\,j)}{\left[\mathbf{Q}_-^{(t)}\mathbf{A}^{(t-1)}\mathbf{Q}_+^{(t)}\right]_{ij}}. \quad (15)$$

In Gaussier and Goutte (2005), Ding et al. (2008), Shashanka et al. (2007, 2008), and Smaragdis and Raj (2007) the close relationship between NMF with the KL divergence, PLSA and other PLVM approaches was commented. In Gaussier and Goutte (2005), it was shown how NMF problems may be interpreted as PLSA (Hofmann 1999). In Ding et al. (2008) the authors showed that NMF and PLSA even though they minimize similar objective functions they do not converge to the same limit point. Moreover, in Shashanka et al. (2007, 2008) the relationship between PLVM and NMF algorithms was further studied. Here we shall try to provide an interpretation optimization problem (6) and update rules (8), (9) and (10) in terms of PLVM. Moreover, we will show that the constrained optimization problem (12) which is solved via update rules (13), (14) and (15) is exactly the same as PLSA algorithm in Hofmann (1999).

Latent class models, like PLSA, enable one to attribute the observations to hidden or latent factors. The main characteristic of these models is that conditionally independent bivariate data are modelled as belonging to latent classes, such that the random variables within a latent class are independent of one another. Random variables (X_1, X_2) can be thought of as if they were defined in the canonical measurable space $(\Omega, \mathcal{F})$, where Ω is the set of all pairs (i_1, i_2) and $\mathcal{F} = 2^{\Omega}$ (i.e., the powerset of Ω). Let $\mathbb{P}$ be a probabilistic measure on this space. Then, the distribution of the pair (X_1, X_2) under $\mathbb{P}$ is given by matrix $\mathbf{P} = [p_{ij}]$. We shall use the notation $P(x_1, x_2)$ for distribution $\mathbb{P}(x_1 = i, x_2 = j) = p_{ij}$. Distribution $P(x_1, x_2)$ is modelled as a mixture, where each component of the mixture is the product of one-dimensional marginal distributions. Let x_1 and x_2 be random variables and let $P(x_1, x_2)$ be their joint probability mass function. The PLSA approximation may be written as:

$$P(x_1, x_2) \approx \sum_{z \in \{1,\ldots,K\}} P(x_1, z) P(x_2|z) \tag{16}$$

where z is a latent random variable. In an EM (Hofmann 1999) (EM) approach we maximize the following functional:

$$\begin{aligned} &D_{\text{PLSA}}\left(\mathbb{P}(x_1, x_2)||\mathbb{P}(x_1, z)\mathbb{P}(x_2|z)\right) \\ &= \sum_{i_1=1, i_2=1}^{I_1, I_2} P(x_1 = i_1, x_2 = i_2) \ln \left(\sum_{i_3=1}^{K} P(x_1 = 1, z = i_3) P(x_2 = i_2|z = i_3) \right) \\ &= -D_{\text{KL}}\left(\mathbb{P}(x_1, x_2)||\mathbb{P}(x_1, z)\mathbb{P}(x_2||z)\right) - c \end{aligned} \tag{17}$$

where c is a constant $c = \sum_{i_1=1, i_2=1}^{I_1, I_2} P(x_1 = i_1, x_2 = i_2) \ln(P(x_1 = i_1, x_2 = i_2))$. Thus, as can be seen is equivalent to minimization of $D_{\text{KL}}\left(\mathbb{P}(x_1, x_2)||\mathbb{P}(x_1, z)\right)$. Solving maximization of (17) (Hofmann 1999) we have the following updating rules:

$$P^{(t)}(z|x_1, x_2) = \frac{P^{(t-1)}(x_1, z) P^{(t-1)}(x_2|z)}{\sum_{y \in \{1,2,\ldots,K\}} P^{(t-1)}(x_1, y) P^{(t-1)}(x_2|y)} \tag{18}$$

and

$$P^{(t)}(x_1, z) = \sum_{a_2} P(x_1, a_2) P^{(t)}(z|x_1, a_2)$$
$$P^{(t)}(x_2|z) = \frac{\sum_{a_1} P(a_1, x_2) P^{(t)}(z|a_1, x_2)}{\sum_{a_1,a_2} P(a_1, a_2) P^{(t)}(z|a_1, a_2)}. \tag{19}$$

Now by substituting (18) into (19) we have:

$$\begin{aligned} P^{(t)}(x_1, z) &= P^{(t-1)}(x_1, z) \sum_{a_2} P(x_1, a_2) \frac{P^{(t-1)}(a_2|z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t-1)}(x_1, y) P^{(t-1)}(a_2|y)} \\ P^{(t)}(x_2|z) &= P^{(t-1)}(x_2|z) \frac{\sum_{a_1} P(a_1, x_2) \frac{P^{(t)}(a_1,z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t)}(a_1,y) P^{(t-1)}(x_2|y)}}{\sum_{a_1,a_2} P(a_1, a_2) \frac{P^{(t)}(a_1,z) P^{(t-1)}(a_2|z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t)}(a_1,y) P^{(t-1)}(a_2|y)}} \\ &= P^{(t-1)}(x_2|z) \frac{\sum_{a_1} P(a_1, x_2) \frac{P^{(t)}(a_1,z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t)}(a_1,y) P^{(t-1)}(x_2|y)}}{\sum_{a_1} P^{(t)}(a_1, z)}. \end{aligned} \tag{20}$$

Then, since $\sum_{i=1}^{F} \sum_{k=1}^{M} \mathbb{P}(x_1 = i, z = k) = 1$ and $\sum_{j=1}^{L} \mathbb{P}(x_2 = j|z = k) = 1$ (which are equivalent to $\mathbf{e}^T \mathbf{Q}_- \mathbf{e} = 1$ and $\mathbf{Q}_+ \mathbf{e} = \mathbf{e}$, respectively) we can see that $\mathbf{Q}_+(i, k) = \mathbb{P}(x_1 = i, z = k)$ and $[\mathbf{Q}_-]^T(j, k) = \mathbb{P}(x_2 = j, z = k)$. Thus, the update rules (8), (9) and (10) are exactly the same as the EM algorithm (20).

In case that we that want to calculate the probabilities $P(z)$ [which is computed in the PLSA algorithm in Hofmann (1999)] as well then approximation (21) is reformulated as:

$$P(x_1, x_2) \approx \sum_{z\in\{1,\ldots,K\}} P(z) P(x_1|z) P(x_2|z) \tag{21}$$

and optimization problem is the maximization of $D_{\text{PLSA}}(\mathbb{P}(x_1, x_2)||\mathbb{P}(x_1, z)\mathbf{diag}(\mathbb{P}(z))\mathbb{P}(x_2|z))$ which is the PLSA model proposed in Hofmann (1999). The corresponding update rules are given by:

$$\begin{aligned} P^{(t)}(x_1|z) &= P^{(t-1)}(x_1|z) \frac{\sum_{a_2} P(x_1, a_2) \frac{P^{(t-1)}(z) P^{(t-1)}(a_2|z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t-1)}(y) P^{(t-1)}(x_1|y) P^{(t-1)}(a_2|y)}}{\sum_{a_1,a_2} P(a_1, a_2) \frac{P^{(t-1)}(z) P^{(t-1)}(a_2|z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t-1)}(y) P^{(t-1)}(a_1|y) P^{(t-1)}(a_2|y)}} \\ P^{(t)}(x_2|z) &= P^{(t-1)}(x_2|z) \frac{\sum_{a_1} P(a_1, x_2) \frac{P^{(t-1)}(z) P^{(t)}(a_1|z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t-1)}(y) P^{(t)}(a_1|y) P^{(t-1)}(x_2|y)}}{\sum_{a_1,a_2} P(a_1, a_2) \frac{P^{(t-1)}(z) P^{(t)}(a_1|z) P^{(t-1)}(a_2|z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t-1)}(y) P^{(t)}(a_1|y) P^{(t-1)}(a_2|y)}} \end{aligned} \tag{22}$$

and

$$P^{(t)}(z) = P^{(t-1)}(z) \sum_{a_1,a_2} \frac{P(a_1,a_2)P^{(t)}(a_1|z)P^{(t)}(a_2|z)}{\sum_{y\in\{1,2,\ldots,K\}} P^{(t-1)}(y)P^{(t)}(a_1|y)P^{(t)}(a_2|y)}. \quad (23)$$

We observe that since $\sum_{i=1}^{F} \mathbb{P}(x_1 = i|z = k) = 1$, $\sum_{j=1}^{L} \mathbb{P}(x_2 = j|z = k) = 1$ and $\sum_{k=1}^{K} P(z = k)$ (which are equivalent to $\mathbf{Q}_-^T\mathbf{e} = 1$, $\mathbf{Q}_+\mathbf{e} = \mathbf{e}$ and $\mathbf{a}^T\mathbf{e} = 1$, respectively) we can see that $\mathbf{Q}_-(i,k) = \mathbb{P}(x_1 = i|z = k)$, $[\mathbf{Q}_+]^T(j,k) = \mathbb{P}(x_2 = j|z = k)$ and $\mathbf{a}(k) = \mathbb{P}(z = k)$. Thus, update rules (8), (9) and (10) are exactly the same as the PLSA algorithm given by (22) and (23).

2.3 Relationship between PLSA, original NMF and the approach proposed in Ding et al. (2008)

The relationship between NMF and PLSA has been commented in Gaussier and Goutte (2005), Ding et al. (2008), and Shashanka et al. (2008). Now, we will discuss the actual difference between NMF and PLSA and complete the analysis initiated in Ding et al. (2008). In Ding et al. (2008) the authors proposed an algorithm for solving the approximation (16) using the actual NMF update rules (4) and (5). After the computation of $\mathbf{Z}$ and $\mathbf{H}$ they computed matrices $\mathbf{Q}_-(i,k) = \mathbb{P}(x_1 = i|z = k)$, $\mathbf{Q}_+(k,j) = \mathbb{P}(x_2 = j|z = k)$ and $\mathbf{a}(k) = \mathbb{P}(z = k)$ (the diagonal of $\mathbf{A}$):

$$\begin{aligned} \mathbf{Q}_- &= \frac{\mathbf{Z}}{\mathbf{EZ}} \\ \mathbf{Q}_+ &= \frac{\mathbf{H}}{\mathbf{HE}} \\ \mathbf{A} &= \mathbf{diag}\left(\mathbf{e}^T\mathbf{Z}\right)\mathbf{diag}(\mathbf{He}). \end{aligned} \quad (24)$$

As the authors showed the above procedure results to a different limit point than the limit point derived from update rules (22) and (23). What the authors actually showed in Ding et al. (2008) is that the algorithm will converge to a different limit point when the factors are normalized under every iteration such that the constraints $\mathbf{Q}_-^T\mathbf{e} = 1$, $\mathbf{Q}_+\mathbf{e} = \mathbf{e}$ and $\mathbf{a}^T\mathbf{e} = 1$ are satisfied and it will converge to another limit point when the normalization is applied once at the end of the procedure (i.e., at the convergence). This is quite expected since these are two different optimization problems. The first one is the NMF optimization problem (4) having free variables $\mathbf{Z}$ and $\mathbf{H}$ which solution is given by update rules (4) and (5). The second one is the constrained optimization problem (12) having as free variables $\mathbf{Q}_-$, $\mathbf{Q}_+$, $\mathbf{a}$ (subject to the additional constraints $\mathbf{Q}_-^T\mathbf{e} = \mathbf{e}$, $\mathbf{Q}_+\mathbf{e} = \mathbf{e}$ and $\mathbf{a}^T\mathbf{e} = 1$) which solution is computed by update rules (13), (14) and (15). A similar analysis also hold for the optimization problem (6). That is, even though (12) and (6) are similar they are not equivalent. Thus, they do not converge to the same limit point.

Summarizing, we provided a supplement to the analysis given in Ding et al. (2008) and we showed that even though optimization problems (4), (6) and (12) optimize

similar objective function they are not equivalent, since they are optimization problems with different set of variables (for example in PLSA we search for an additional vector **a**) and with different constraints (and thus a different search space). Only the constrained NMF given by optimization problem (12) and solved by update rules (13), (14) and (15) is equivalent to PLSA and both converge to the same limit point.

3 Nonnegative Tensor Factorization

In this section we shall briefly describe elements of multilinear algebra and show how to perform nonnegative tensor factorizations.

3.1 Tensor models and PARAFAC decomposition

To begin with, let us briefly review the tensor algebra concepts needed hereafter and define the notation that will be used throughout the paper. An n-order tensor is a collection of measurements indexed by n indices, where each index refers to a mode. Accordingly, vectors are first-order tensors and matrices are second-order tensors (Kolda and Bader 2009). Lower case letters (e.g. x) have already been used to denote scalars in the introduction, while boldface lowercase letters (e.g. $\mathbf{x}$) and boldface capital letters (e.g. $\mathbf{X}$) have denoted vectors and matrices, respectively. Higher-order tensors (of order 3 or higher) are denoted by boldface Euler script calligraphic letters (e.g. $\boldsymbol{\mathcal{X}}$). If the ith element of a vector $\mathbf{x} \in \mathbb{R}_+^I$ is denoted by x_i, $i = 1, 2, \ldots, I$, then the elements of an n-order tensor $\boldsymbol{\mathcal{X}}$ will be denoted by $x_{i_1 i_2 \ldots i_n}$, $i_\ell = 1, 2, \ldots, I_\ell$, $\ell = 1, 2, \ldots, n$. In the following, we shall focus on nth order tensors with non-negative elements, i.e. $\boldsymbol{\mathcal{X}} \in \mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_n}$. For example:

- $\boldsymbol{\mathcal{X}}$ could be the representation of a database with L objects. Then every database object is a nonnegative tensor of order $(n-1)$ denoted as $\boldsymbol{\mathcal{X}}_{:i_n} \in \mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_{n-1}}$, $i_n = 1, 2, \ldots, L$, that is indexed by an $(n-1)$ tuple of indices $(i_1, i_2, \ldots, i_{n-1})$. To be more specific, the most natural way to model a facial image database is by a 3rd order tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}_+^{I_1 \times I_2 \times I_3}$, where I_1 and I_2 refer to the image hight and width, respectively and $I_3 = L$ is the number of images in the database (Shashua and Hazan 2005).
- A relationship that refers to more than two modes is more naturally represented by a tensor $\mathcal{X}$. Such relationships could be nonnegative affinity measures over an n-tuple of patterns that frequently arise in visual interpretation problems, including 3D multi-body segmentation and illumination-based clustering of human faces (Shashua and Hazan 2005).
- An n-variate distribution is more naturally modelled as an n-order tensor.

Frequently, transformations of tensors into matrices (ℓ-mode matricization) and matrices into vectors (vectorization) are needed. The mode-ℓ matricization of a tensor $\boldsymbol{\mathcal{X}} \in \mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_n}$ maps $\boldsymbol{\mathcal{X}}$ to a matrix $\mathbf{X}_{(\ell)} \in \mathbb{R}^{I_\ell \times M}$ with $M = \prod_{\substack{m=1 \\ m \neq \ell}}^{n} I_m$ such that the tensor element $x_{i_1 i_2 \ldots i_n}$ is mapped to the matrix element $x_{i_\ell j}$ where $j = 1 + \sum_{\substack{k=1 \\ k \neq \ell}}^{n} (i_k - 1) J_k$ with $J_k = \prod_{\substack{m=1 \\ m \neq \ell}}^{k-1} I_m$. The operator **vec**() applied to a matrix

stacks the matrix columns into a single vector and the operator. The matrix of ones and the identity matrix are denoted by $\mathbf{E}$ and $\mathbf{I}$, respectively. A tensor of ones is similarly denoted as $\mathcal{E}$. A tensor which has all elements equal to 0 except those for which all indices are the same is called superdiagonal. If the nonzero elements are equal to 1, then such a superdiagonal tensor is referred to as the unit superdiagonal tensor $\mathfrak{J}$. In the remaining of the paper, $\bullet$ and / tensor operators will denote the elementwise multiplication and division between tensors.

In the following, we shall need several products between vectors and matrices as well as between tensors and matrices. Let $\mathbf{a} \in \mathbb{R}_+^I$ and $\mathbf{b} \in \mathbb{R}_+^J$ be two non-negative real valued vectors. Their outer product yields a matrix $\mathbf{C} \in \mathbb{R}_+^{I \times J}$

$$\mathbf{C} = \mathbf{a} \circ \mathbf{b} \text{ with elements } c_{ij} = a_i \, b_j. \tag{25}$$

Consequently, the outer product of n vectors $\mathbf{a}_\ell \in \mathbb{R}_+^{I_\ell}$, $\ell = 1, 2, \ldots, n$, $\mathbf{a}_1 \circ \mathbf{a}_2 \circ \cdots \circ \mathbf{a}_n = \bigcirc_{\ell=1}^n \mathbf{a}_\ell$ yields a tensor $\mathcal{A} \in \mathbb{R}_+^{I_1 \times I_2 \times \cdots \times I_n}$.

The Kronecker product between two matrices $\mathbf{A} \in \mathbb{R}_+^{I_1 \times M_1}$ and $\mathbf{B} \in \mathbb{R}_+^{I_2 \times M_2}$ is defined as:

$$\begin{aligned} \mathbf{A} \otimes \mathbf{B} &= \begin{bmatrix} a_{11}\mathbf{B} & a_{12}\mathbf{B} \cdots & a_{1M_1}\mathbf{B} \\ a_{21}\mathbf{B} & a_{22}\mathbf{B} \cdots & a_{2M_1}\mathbf{B} \\ \vdots & \vdots & \ddots \quad \vdots \\ a_{I_1 1}\mathbf{B} & a_{I_1 2}\mathbf{B} \cdots & a_{I_1 M_1}\mathbf{B} \end{bmatrix} \\ &= \left[\mathbf{a}_1 \otimes \mathbf{b}_1 \; \mathbf{a}_1 \otimes \mathbf{b}_2 \ldots \mathbf{a}_1 \otimes \mathbf{b}_{M_2} \mathbf{a}_2 \otimes \mathbf{b}_1 \ldots \mathbf{a}_{M_1} \otimes \mathbf{b}_1 \ldots \mathbf{a}_{M_1} \otimes \mathbf{b}_{M_2}\right] \end{aligned} \tag{26}$$

and yields a nonnegative matrix of size $I_1 \, I_2 \times M_1 \, M_2$. Accordingly, the Kronecker product of n matrices $\mathbf{A}_\ell \in \mathbb{R}_+^{I_i \times M_i}$, $\ell = 1, 2, \ldots, n$, denoted compactly as $\bigotimes_{\ell=1}^n \mathbf{A}_\ell$ yields a matrix of size $\prod_{\ell=1}^n I_i \times \prod_{\ell=1}^n M_i$.

The Khatri-Rao product between two matrices $\mathbf{A} = [\mathbf{a}_1 | \mathbf{a}_2 | \ldots | \mathbf{a}_K] \in \mathbb{R}_+^{I \times K}$ and $\mathbf{B} = [\mathbf{b}_1 | \mathbf{b}_2 | \ldots | \mathbf{b}_K] \in \mathbb{R}_+^{J \times K}$ results in a matrix of size $(I \, J) \times K$. It is defined as the matching columnwise Kronecker product of the aforementioned matrices, i.e.

$$\mathbf{A} \odot \mathbf{B} = [\mathbf{a}_1 \otimes \mathbf{b}_1 | \mathbf{a}_2 \otimes \mathbf{b}_2 | \ldots | \mathbf{a}_K \otimes \mathbf{b}_K]. \tag{27}$$

If we have n matrices $\mathbf{A}_\ell \in \mathbb{R}_+^{I_\ell \times K}$, $\ell = 1, 2, \ldots, n$, their Khatri-Rao product is compactly denoted as $\bigodot_{\ell=1}^n \mathbf{A}_\ell$ and yields a nonnegative matrix of size $\left(\prod_{\ell=1}^n I_\ell\right) \times K$.

The two most commonly used tensor decompositions are the PARAFAC/CANDECOMP model (Harshman 1970; Carroll and Chang 1970) [which is also equivalent to the decomposition using 1-order Kruskal tensors (Kruskal 1977)] and the Tucker tensor models (Tucker 1966). Another way to perform the decomposition, especially for 3D tensor factorization, is given through the PARAFAC2 model (Smilde et al. 2004) which has been applied for 3D NTF in Cichocki et al. (2007). An n-order tensor $\mathcal{X}$ is of rank at most K if it can be expressed as the sum of K rank-1 Kruskal tensors

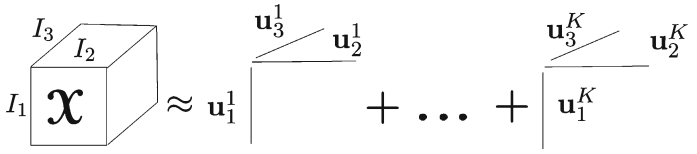

Fig. 1 Visualization of the rank-K approximation of a 3-order tensor using Kruskal tensor notation

i.e., a sum of Kn-fold outer-products. Using the PARAFAC tensor model[1] a tensor $\mathcal{X}$ can be decomposed as the sum of Kn-fold outer-products as:

$$\begin{aligned} \mathcal{X} &\approx \sum\nolimits_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l \Leftrightarrow \\ x_{i_1 \ldots i_n} &\approx \sum\nolimits_{l=1}^{K} u_{i_1 1}^l \ldots u_{i_n n}^l, \ 0 \le i_j \le I_j, \ 1 \le j \le n \end{aligned} \tag{28}$$

with $\mathbf{u}_j^l \in \Re_+^{I_j}$ and $\mathbf{u}_j^l = \left[u_{1j}^l, \ldots, u_{I_j j}^l\right]^T$. That is, NTF aims at finding the best rank K nonnegative approximation of $\mathcal{X}$ with respect to an approximation cost. The NTF factorization using Kruskal tensors is pictorially described in Fig. 1. As can be seen, tensor $\mathcal{X} \in \Re_+^{I_1 \times I_2 \times I_3}$ is represented as the sum of K outer product tensors $\mathbf{u}_1^l \circ \mathbf{u}_2^l \circ \mathbf{u}_3^l$.

Decompositions like (28) can be calculated using only matrix operations. To do so, let us define matrices $\mathbf{U}_j \triangleq \left[\mathbf{u}_j^1 | \ldots | \mathbf{u}_j^K\right]$ which contain as columns the Kruskal vectors, $j = 1, \ldots, n$ or equivalently $\mathbf{U}_j = \left[u_{i_j j}^l\right] \in \Re_+^{I_j \times K}$, $1 \le i_j \le I_j$, $1 \le j \le n$, $1 \le l \le n$. These matrices will be used for defining NTF algorithms using matrix multiplications.

Let us introduce the compact notation

$$\overline{\bigodot}_j \mathbf{U}_l \triangleq \mathbf{U}_n \odot \cdots \odot \mathbf{U}_{j+1} \odot \mathbf{U}_{j-1} \odot \cdots \odot \mathbf{U}_1. \tag{29}$$

Using compact notation (29) the nonnegative factorization (28) can be written in a matricized form as:

$$\mathbf{X}_{(j)} \approx \mathbf{R}_{(j)} = \mathbf{U}_j \mathbf{Z}_{(j)}^T = \mathbf{U}_j \left(\overline{\bigodot}_j \mathbf{U}_l\right)^T \tag{30}$$

matrices $\mathbf{R}_{(j)} \in \Re_+^{I_j \times \prod_{i=1, i \ne j}^{n} I_i}$ and $\mathbf{Z}_{(j)} \in \Re_+^{\prod_{i=1, i \ne j}^{n} I_i \times K}$ where $\mathbf{Z}_{(j)} = \overline{\bigodot}_j \mathbf{U}_i$ will be helpful in defining algorithms for the above factorization. The NMF problem $\mathbf{X} \approx \mathbf{ZH}$ or $\mathbf{x}_j \approx \mathbf{Z}\mathbf{h}_j$ in Lee and Seung (2000) can be easily derived from (28) by selecting $\mathbf{Z} = \mathbf{U}_1$ and $\mathbf{H} = \mathbf{U}_2^T$.

[1] In the rest of the paper PARAFAC tensor decomposition and decomposition using the Kruskal tensor model will be used interchangeably in order to denote the same tensor decomposition.

3.2 Nonnegative Tensor Factorization algorithms for KL-divergence

The first method for NTF was proposed in Paatero and Tapper (1994). The NMF methods proposed in Lee and Seung (2000) were generalized using PARAFAC tensor decomposition in Shashua and Hazan (2005). Recently, in Kim and Choi (2007) and Mørup et al. (2008) NTF methods using Tucker decomposition were proposed. Moreover, in Cichocki et al. (2008) PAFARAC2 models for 3D NTF were also proposed.

In Hazan et al. (2005) the authors proposed an NTF algorithm using the KL-divergence for 3-order tensors. NTF with the KL-divergence for arbitrary order tensors was proposed in Zafeiriou (2009a). In the general n-order tensor case, the KL divergence between the given tensor $\mathcal{X}$ and the sum of rank-1 tensors is:

$$D_{\mathrm{KL}}\left(\mathcal{X}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{u}_j^l\right)=\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}\left(x_{i_1\ldots i_n}\ln\left(\frac{x_{i_1\ldots i_n}}{\sum_{m=1}^{K}\prod_{j=1}^{n}u_{i_j j}^m}\right)-x_{i_1\ldots i_n}+\sum_{m=1}^{K}\prod_{j=1}^{n}u_{i_j j}^m\right). \tag{31}$$

The optimization problem of this NTF decomposition is:

$$\begin{aligned}&\min_{\mathbf{u}_{i_j}^m} D_{\mathrm{KL}}\left(\mathcal{X}||\sum\nolimits_{m=1}^{K}\bigcirc_{j=1}^{n}\mathbf{u}_j^m\right)\ \text{subject to}\\ &u_{i_j j}^m\geq 0,\ \forall\ 1\leq i_j\leq I_j, 1\leq j\leq n, 1\leq m\leq K\end{aligned} \tag{32}$$

The solution of the above optimization problem was calculated via the use of an auxiliary function. W is an auxiliary function of $Y(\mathbf{F})$ if $W\left(\mathbf{F},\mathbf{F}^{(t-1)}\right)\geq Y(\mathbf{F})$ and $W(\mathbf{F},\mathbf{F})=Y(\mathbf{F})$. If W is an auxiliary function of Y, then Y is nonincreasing under the update $\mathbf{F}^{(t)}=\arg\min_{\mathbf{F}}W\left(\mathbf{F},\mathbf{F}^{(t-1)}\right)$ (Lee and Seung 2000). With the help of the auxiliary function, the update rules for $\mathbf{U}_j$ can be derived. By fixing matrices $\mathbf{U}_1,\ldots,\mathbf{U}_{j-1},\mathbf{U}_{j+1},\ldots,\mathbf{U}_n$, the elements of matrix $\mathbf{U}_j$ are updated by minimizing $Y(\mathbf{U}_j)=D_{\mathrm{KL}}\left(\mathcal{X}||\sum_{m=1}^{K}\bigcirc_{j=1}^{n}\mathbf{u}_j^m\right)=D_{\mathrm{KL}}\left(\mathbf{X}_{(j)}||\mathbf{U}_j\left(\overline{\odot_j}\mathbf{U}_l\right)^T\right)$. For matrix $\mathbf{U}_j$ we define function:

$$\begin{aligned}W(\mathbf{U}_j^{(t)},\mathbf{U}_j^{(t-1)})&\triangleq\sum\nolimits_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}(x_{i_1\ldots i_n}\ln(x_{i_1\ldots i_n})-x_{i_1\ldots i_n})\\ &+\sum\nolimits_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}x_{i_1\ldots i_n}\sum\nolimits_{m=1}^{K}\frac{u_{i_j j}^{m\ (t-1)}\prod_{\substack{r=1\\r\neq j}}^{n}u_{i_r r}^m}{\sum_{l=1}^{K}u_{i_j j}^{l\ (t-1)}\prod_{\substack{r=1\\r\neq j}}^{n}u_{i_r r}^l}\left(\ln\left(u_{i_j j}^{l\ (t)}\prod_{\substack{r=1\\r\neq j}}^{n}u_{i_r r}^l\right)\right.\\ &\left.-\ln\frac{u_{i_j j}^{m\ (t-1)}\prod_{\substack{r=1\\r\neq j}}^{n}u_{i_r r}^m}{\sum_{l=1}^{K}u_{i_j j}^{l\ (t-1)}\prod_{\substack{r=1\\r\neq j}}^{n}u_{i_r r}^l}\right)+\sum\nolimits_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}\sum\nolimits_{m=1}^{K}u_{i_j j}^{m\ (t)}\prod_{\substack{r=1\\r\neq j}}^{n}u_{i_r r}^m.\end{aligned} \tag{33}$$

Function (33) is an auxiliary function for (31). A proof of the above statements is given in Zafeiriou (2009a).

Let us define the compact notation

$$\overline{\sum_{i_j}} \triangleq \sum_{i_1=1}^{I_1} \sum_{i_2=1}^{I_2} \cdots \sum_{i_{j-1}=1}^{I_{j-1}} \sum_{i_{j+1}=1}^{I_{j+1}} \cdots \sum_{i_n=1}^{I_n}. \tag{34}$$

and

$$\begin{aligned} \prod_{\substack{r=1 \\ (j,t)}}^{n} u_{i_r,r}^{m} &\triangleq u_{i_j j}^{m\ (t-1)} \prod_{\substack{r=1, r\neq j \\ (j,t)}}^{n} u_{i_r r}^{m} \\ &\triangleq u_{i_j j}^{m\ (t-1)} u_{i_1 1}^{m\ (t)} \cdots u_{i_{j-1} j-1}^{m\ (t)} u_{i_{j+1} j+1}^{m\ (t-1)} \cdots u_{i_n n}^{m\ (t-1)}. \end{aligned} \tag{35}$$

The above calculates the product $\prod_{r=1}^{n} u_{i_r r}^{m}$ where for $r = 1, \ldots, j-1$ we use the estimate of $u_{i_r r}^{m}$ at time t while for the rest of them we use the estimate at time $t-1$.

Having defined the above compact notations the update rules that can guarantee a non increasing behavior of cost function (31) for factors $u_{i_j j}^{m\ (t)}$, are obtained by solving $\frac{\partial W\left(\mathbf{U}_j^{(t)}, \mathbf{U}_j^{(t-1)}\right)}{\partial u_{i_j j}^{m}} = 0$ and are given by:

$$u_{i_j j}^{m\ (t)} = u_{i_j j}^{m\ (t-1)} \frac{\overline{\sum_{i_j}} x_{i_1 \ldots i_n} \frac{\prod_{\substack{r=1, r\neq j \\ (j,t)}}^{n} u_{i_j j}^{m}}{\sum_{l=1}^{K} \prod_{\substack{r=1 \\ (j,t)}}^{n} u_{i_r r}^{l}}}{\overline{\sum_{i_j}} \prod_{\substack{r=1 \\ (j,t)}}^{n} u_{i_j j}^{m}}. \tag{36}$$

The above update rules can be formulated using matrix operations. To do so, let us define notation

$$\mathbf{Z}_{(j)}^{(t-1)} \triangleq \mathbf{U}_n^{(t-1)} \odot \cdots \odot \mathbf{U}_{j+1}^{(t-1)} \odot \mathbf{U}_{j-1}^{(t)} \odot \cdots \odot \mathbf{U}_1^{(t)} \tag{37}$$

and

$$\mathbf{R}_{(j)}^{(t-1)} \triangleq \mathbf{U}_j^{(t-1)} \left[\mathbf{Z}_{(j)}^{(t-1)}\right]^T. \tag{38}$$

Using the above definitions, the update rules (36) are given by:

$$\mathbf{U}_j^{(t)} = \mathbf{U}_j^{(t-1)} \bullet \left(\frac{\mathbf{X}_{(j)} \mathbf{Z}_{(j)}^{(t-1)T}}{\mathbf{R}_{(j)}^{(t-1)} \mathbf{E} \mathbf{Z}_{(j)}^{(t-1)}} \right). \tag{39}$$

If we allow a total of r iterations, the complexity of the NTF algorithm is $O\left(rnK \prod_{j=1}^{n} I_j\right)$.

In Sra and Dhillon (2006) exponential rules for NMF using the KL divergence were proposed. The exponential rules for arbitrary valence NTF were proposed in Zafeiriou (2009a). For the exponential rules the following problem was considered:

$$\min_{\mathbf{u}_{i_j}^m} D_{\mathrm{KL}}\left(\sum\nolimits_{m=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^m || \mathcal{X}\right) \text{ subject to}$$
$$u_{i_j j}^m \geq 0, \ \forall\ 1 \leq i_j \leq I_j, 1 \leq j \leq n, 1 \leq m \leq K. \tag{40}$$

The update rules that guarantee a nonincreasing behavior of the above cost function are the following (Zafeiriou 2009a):

$$u_{i_j j}^{l\ (t)} = u_{i_j j}^{l\ (t-1)} \exp\left(\frac{\overline{\sum_{i_j}} \prod_{\substack{r=1, r\neq j \\ (j,t)}}^{n} u_{i_r r}^{l} \ln \frac{x_{i_1 \ldots i_n}}{\sum_{m=1}^{n} \prod_{\substack{r=1 \\ (j,t)}}^{n} u_{i_r r}^{m}}}{\overline{\sum_{i_j}} \prod_{\substack{r=1 \\ (j,t)}}^{n} u_{i_r r}^{l}}\right). \tag{41}$$

or in matrix notation:

$$\mathbf{U}_j^{(t)} = \mathbf{U}_j^{(t-1)} \bullet \exp\left(\frac{\ln\left(\frac{\mathbf{X}_{(j)}}{\mathbf{R}_{(j)}^{(t-1)}}\right) \mathbf{Z}_j^{(t-1)T}}{\mathbf{E}\mathbf{Z}_{(j)}^{\ (t-1)}}\right). \tag{42}$$

4 NTF with KL divergence as an alternative Csiszar–Tusnady procedure and convergence properties

In this section we will formally define the proposed NTF with KL divergence, provide a clear probabilistic interpretation and explore its convergence properties. First, let us consider the optimization problem with constraints:

$$\min_{\mathbf{U}_1, \ldots, \mathbf{U}_n : \mathbf{e}^T \mathbf{U}_1 \mathbf{e} = 1, \mathbf{U}_2{}^T \mathbf{e} = \mathbf{e}, \ldots, \mathbf{U}_n{}^T \mathbf{e} = \mathbf{e}} D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right). \tag{43}$$

Proposition 2.1 *The NTF optimization problem* (43) *has a solution.*

The proof of the above proposition can be found in Appendix 10.

Before starting to explore the properties of NTF with KL Divergence, we should try to find the resemblance between the robust NMF method proposed in Finesso and Spreij (2006) and NTF decomposition. NMF is a special case of NTF when having 2-order tensors (i.e. having only $\mathbf{U}_1$ and $\mathbf{U}_2$), thus the NTF optimization problem is exactly the one in Finesso and Spreij (2006) under the constraints $\mathbf{e}^T \mathbf{U}_1 \mathbf{e} = 1$ and $\mathbf{U}_2^T \mathbf{e} = \mathbf{e}$. The generalized optimization problem for NTF using KL divergence is the one expressed by (43).

In order to define the robust NTF algorithm we should reformulate the optimization problem using probability matrices. That is, we shall define the probability tensor as

$\mathcal{P}$ such that $p_{i_1 i_2 \ldots i_n} \geq 0$ and $\sum_{i_1=1,i_2=1,\ldots,i_n=1}^{I_1,I_2,\ldots,I_n} p_{i_1 i_2 \ldots i_n} = 1$. For convenience, we shall use the notation $\mathcal{P}(i_1, \ldots, i_n)$ interchangeably with $p_{i_1 \ldots i_n}$.

Let $p \triangleq \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} x_{i_1 \ldots i_n}$ and $w_1 \triangleq \mathbf{e}^T \mathbf{U}_1 \mathbf{e}$ then we define $\mathcal{P} \triangleq \frac{1}{p}\mathcal{X}$, $\mathbf{Q}_1 \triangleq \frac{1}{w_1}\mathbf{U}_1$ and $\mathbf{Q}_i \triangleq \mathbf{U}_i$ for all $i = 2, \ldots, n$. As proven in Appendix 11 cost function $D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right)$ can be reformulated as:

$$D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right) = p D_{\mathrm{KL}}\left(\mathcal{P} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{q}_j^l\right) + D_{\mathrm{KL}}(p||w_1). \quad (44)$$

Since number p is known, the minimization of $D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right)$ with respect to $(\mathbf{U}_1, \ldots, \mathbf{U}_n)$ is equivalent to minimizing $D_{\mathrm{KL}}\left(\mathcal{P} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{q}_j^l\right)$ with respect to $\mathbf{Q}_1$,…,$\mathbf{Q}_n$ and minimizing $D_{\mathrm{KL}}(p||w_1)$ with respect to w_1. For the new optimization problem, we obtain $w_1^{(\mathrm{opt})} = p$ and so $\mathbf{U}_1^{(\mathrm{opt})} = p\mathbf{Q}_1^{(\mathrm{opt})}$ and $\mathbf{U}_j^{(\mathrm{opt})} = \mathbf{Q}_j^{(\mathrm{opt})}$, $j = 2, \ldots, n$.[2] Thus, minimization of $D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right)$ subject to nonnegativity constraints is equivalent to minimizing $D_{\mathrm{KL}}\left(\mathcal{P} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{q}_j^l\right)$ subject to nonnegativity constraints. This leads to the minimization of the KL-Divergence between (finite) probability measures:

$$\begin{aligned} D_{\mathrm{KL}}\left(\mathcal{P} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{q}_j^l\right) &= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} D_{\mathrm{KL}}\left(p_{i_1,\ldots,i_n} || \sum_{l=1}^{K} \prod_{j=1}^{n} q_{i_j j}^l\right) \\ &= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} p_{i_1,\ldots,i_n} \log\left(\frac{p_{i_1,\ldots,i_n}}{\sum_{l=1}^{K} \prod_{j=1}^{n} q_{i_j j}^l}\right). \end{aligned} \quad (45)$$

Given a probability tensor $\mathcal{P}$ and an integer K, the optimization problem is to find $\mathbf{Q}_1, \ldots, \mathbf{Q}_n$:

$$\min_{\mathbf{Q}_1,\ldots,\mathbf{Q}_n : \mathbf{e}^T \mathbf{Q}_1 \mathbf{e}=1, \mathbf{Q}_2{}^T \mathbf{e}=\mathbf{e}, \ldots, \mathbf{Q}_n{}^T \mathbf{e}=\mathbf{e}} D_{\mathrm{KL}}\left(\mathcal{P} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{q}_j^l\right). \quad (46)$$

[2] The superscript opt reflects the optimality.

We define the function:

$$W\left(\mathbf{U}_j^{(t)}, \mathbf{U}_j^{(t-1)}\right) = \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} p_{i_1\ldots i_n} \ln(p_{i_1\ldots i_n}) + \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} p_{i_1\ldots i_n} \sum_{m=1}^{K} \frac{q_{ijj}^{m\ (t-1)} \prod_{\substack{r=1\\ r\neq j}}^{n} q_{irr}}{\sum_{l=1}^{K} q_{ijj}^{l\ (t-1)} \prod_{\substack{r=1\\ r\neq j}}^{n} q_{irr}^{l}} \left(\ln\left(q_{ijj}^{l\ (t)} \prod_{\substack{r=1\\ r\neq j}}^{n} q_{irr}^{l} \right) - \ln \frac{q_{ijj}^{m\ (t-1)} \prod_{\substack{r=1\\ r\neq j}}^{n} q_{irr}^{m}}{\sum_{l=1}^{K} q_{ijj}^{l\ (t-1)} \prod_{\substack{r=1\\ r\neq j}}^{l} q_{irr}} \right). \tag{47}$$

This function (47) is an auxiliary function for (45).

Let $\mathbf{Q}_1^{(t)} \triangleq \frac{1}{w_1}\mathbf{U}_1^{(t)}$ and $\mathbf{Q}_2^{(t)} \triangleq \mathbf{U}_2^{(t)}, \ldots, \mathbf{Q}_n^{(t)} \triangleq \mathbf{U}_n^{(t)}$. Now by substituting the definition of $\left(\mathcal{P}, \mathbf{Q}_1{}^{(t)}, \ldots, \mathbf{Q}_n{}^{(t)}\right)$ into (36) and using the fact that $w_1^{(\text{opt})} = p$, the following update rules for $t \geq 1$ are obtained for factors $q_{i_1 1}^{l}$:

$$q_{i_1 1}^{l\ (t)} = q_{i_1 1}^{l\ (t-1)} \overline{\sum_{i_1}} p_{i_1\ldots i_n} \frac{\prod_{\substack{(1,t)\\ j=2}}^{n} q_{ijj}^{l}}{\sum_{m=1}^{K} \prod_{\substack{(1,t)\\ j=1}}^{n} q_{ijj}^{m}} \tag{48}$$

and for $j \in \{2, \ldots, n\}$

$$q_{ijj}^{l\ (t)} = q_{ijj}^{l\ (t-1)} \frac{\overline{\sum_{i_j}} p_{i_1\ldots i_n} \frac{\prod_{\substack{(j,t)\\ r=1, r\neq j}}^{n} q_{irr}{}^{l}}{\sum_{m=1}^{K} \prod_{\substack{(j,t)\\ r=1}}^{n} q_{irr}^{m}}}{\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} p_{i_1\ldots i_n} \frac{\prod_{\substack{(1,t)\\ j=2}}^{n} q_{ijj}^{l}}{\sum_{m=1}^{K} \prod_{\substack{(1,t)\\ j=1}}^{n} q_{ijj}^{m}}}. \tag{49}$$

Update rule (49) can be equivalently written as:

$$q_{ijj}^{l\ (t)} = q_{ijj}^{l\ (t-1)} \frac{\overline{\sum_{i_j}} p_{i_1\ldots i_n} \frac{\prod_{\substack{(j,t)\\ r=1, r\neq j}}^{n} q_{ir,r}^{l}}{\sum_{m=1}^{K} \prod_{\substack{(j,t)\\ r=1}}^{n} q_{ir,r}^{m}}}{\sum_{i_1=1}^{I_1} q_{i_1,1}^{l\ (t)}}. \tag{50}$$

In a matrix notation the above update rules can be written as:

$$\mathbf{Q}_1{}^{(t)} = \mathbf{Q}_j{}^{(t-1)} \bullet \left(\frac{\mathbf{P}_{(1)}}{\mathbf{R}_{(1)}^{(t-1)} \mathbf{Z}_{(1)}^{(t-1)T}} \mathbf{Z}_{(1)}^{(t-1)T} \right) \tag{51}$$

and

$$\mathbf{Q}_j{}^{(t)} = \mathbf{Q}_j{}^{(t-1)} \bullet \frac{\frac{\mathbf{P}_{(j)}}{\mathbf{R}_{(j)}^{(t-1)}\mathbf{Z}_{(j)}^{(t-1)^T}}\mathbf{Z}_{(j)}^{(t-1)^T}}{\mathbf{E}\mathbf{Q}_1^{(t)}} \tag{52}$$

where $\mathbf{Z}_{(j)}^{(t-1)} = \mathbf{Q}_n^{(t-1)} \odot \cdots \odot \mathbf{Q}_{j+1}^{(t-1)} \odot \mathbf{Q}_{j-1}^{(t)} \cdots \odot \mathbf{Q}_1^{(t)}$ and $\mathbf{R}_{(j)}^{(t-1)} = \mathbf{Q}_j^{(t-1)}\mathbf{Z}_{(j)}^{(t-1)^T}$. The following Theorems guarantee that the limit point of the update rules (48) and (49) exists and is a stationary point of optimization problem (46).

Before presenting the convergence theorems we should introduce the following sets.

Given a probability tensor $\mathcal{P}$ and an integer K, we introduce the following sets:

$$\begin{aligned}
\wp &\triangleq \left\{ \mathcal{V} \in \Re_+^{I_1 \times K \times I_2 \times \cdots \times I_n} \ : \ \sum_{l=1}^{K} \mathcal{V}(i_1, l, i_2, \ldots, i_n) = \mathcal{P} \right\} \\
\Im &\triangleq \Big\{ \mathcal{Q} \in \Re_+^{I_1 \times K \times I_2 \times \cdots \times I_n} \ : \ \mathcal{Q}(i_1, l, i_2, \ldots, i_n) = \mathbf{Q}_1(i_1, l)\mathbf{Q}_2(i_2, l) \ldots \mathbf{Q}_n(i_n, l) \\
&\qquad \mathbf{Q}_1, \ldots, \mathbf{Q}_n \geq 0, \qquad \mathbf{e}^T\mathbf{Q}_1\mathbf{e} = 1, \qquad \mathbf{Q}_2^T\mathbf{e} = \mathbf{e}, \ldots, \mathbf{Q}_n^T\mathbf{e} = \mathbf{e} \Big\} \\
\aleph &\triangleq \left\{ \mathcal{O} \in \Re_+^{I_1 \times I_2 \times \cdots \times I_n} \ : \ \mathcal{O} = \sum_{l=1}^{K} \mathcal{Q}(i_1, l, i_2, \ldots, i_n) \text{ for some } \mathcal{Q} \in \Im \right\}
\end{aligned} \tag{53}$$

Theorem 2.1 *Let* $\mathbf{Q}_1^{(t)}, \ldots, \mathbf{Q}_n^{(t)}$ *be generated by algorithm* (48), (49) *and* $\mathcal{Q}^{(t)}$ $(\mathcal{Q}^{(t)}(i_1, l, i_2, \ldots, i_n) = \mathbf{Q}_1^{(t)}(i_1, l)\mathbf{Q}_2^{(t)}(i_2, l) \ldots \mathbf{Q}_n^{(t)}(i_n, l))$ *the corresponding tensor. Then,* $\mathbf{Q}_1^{(t)}(i_1, l)$ *converges to limit* $\mathbf{Q}_1^{\infty}(i_1, l)$ *and* $\mathcal{Q}^{(t)}$ *converges to the limit* $\mathcal{Q}^{\infty}$ *in* $\mathcal{Q}$, $\mathbf{Q}_j^{(t)}(i_j, l)$ *for* $j = 2, \ldots, n$ *converge to limits* $\mathbf{Q}_j^{\infty}(i_j, l)$ *for all* l *with* $\sum_{i_j=1}^{I_j} \mathbf{Q}_j^{\infty}(i_j, l) > 0$.

The proof can be found in Appendix 12. The role of tensor $\mathcal{Q}$ will be made clear in the next section.

Theorem 2.2 *If* $(\mathbf{Q}_1, \ldots, \mathbf{Q}_n)$ *is a limit point of the algorithm given by update rules (48) and (49) in the interior of the domain, then it is a stationary point of the objective function* D_{KL} *(46). If* $(\mathbf{Q}_1, \ldots, \mathbf{Q}_n)$ *is a limit point on the boundary of the domain corresponding to an approximate factorization where none of the columns of* $\mathbf{Q}_1$ *is zero* $(\sum_{i_1=1}^{I_1} \mathbf{Q}_1(i_1, l) > 0 \ \forall l)$*, then all partial derivatives* $\frac{\partial D_{\mathrm{KL}}}{\partial \mathbf{Q}_j(i_j, l)}$ *for* $j = 1, \ldots, n$ *are nonnegative.*

The proof can be found in Appendix 13.

Corollary 2.1 shows that the limit points of the above optimization problem using the update rules (48) and (49) are all Kuhn-Tucker points.

Corollary 2.1 *The limit points of the algorithm with* $\sum_{i_1=1}^{I_1} \mathbf{Q}_1(i_1, l) > 0$ *for all* l *are all the Kuhn-Tucker points for the minimization of* D_{KL} *under the inequality constraints* $\mathbf{Q}_1 \geq 0, \ldots, \mathbf{Q}_n \geq 0$.

The proof of the above Corollary can be found in Appendix 14.

5 A probabilistic interpretation of the optimization problem

In this section we explore the properties of optimization problem (46). More precisely we:

- provide a solid probabilistic interpretation of the optimization problem;
- propose generalized pythagorean rules;
- motivate the use of the auxiliary function (33).

5.1 Setup and exact NTF

Let us consider the $(n+1)$-tuple of random variables $(Y_1, X, Y_2, \ldots, Y_n)$, taking values in $\{1, \ldots, I_1\} \times \{1, \ldots, K\} \times \{1, \ldots, I_2\} \times \cdots \times \{1, \ldots, I_n\}$. The random variables $(Y_1, X, Y_2, \ldots, Y_n)$ can be thought of as if they were defined in the canonical measurable space $(\Omega, \mathcal{F})$, where Ω is the set of all $(n+1)$-ples $(i_1, l, i_2, \ldots, i_n)$ and $\mathcal{F} = 2^{\Omega}$ (i.e., the powerset of Ω). For $\omega = (i_1, l, i_2, \ldots, i_n)$ we have the identity mapping $(Y_1, X, Y_2, \ldots, Y_n)(\omega) = (i_1, l, i_2, \ldots, i_n)$. If $\mathbb{R}$ is a given probabilistic measure on this space, the distribution of the $(n+1)$-tuple $(Y_1, X, Y_2, \ldots, Y_n)$ under $\mathbb{R}$ is given by tensor $\mathcal{R}$:

$$\mathcal{R}(i_1, l, i_2, \ldots, i_n) = \mathbb{R}(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n). \tag{54}$$

Tensors $\mathcal{V}$ and $\mathcal{Q}$ define probability measures $\mathbb{P}$ and $\mathbb{Q}$ on $(\Omega, \mathcal{F})$.

Obviously, the sets $\wp$ and $\Im$ are subsets of the set of all measures on $(\Omega, \mathcal{F})$. In particular, $\wp$ corresponds to the subset of all measures whose $Y = (Y_1, \ldots, Y_n)$ marginal coincides with the given $\mathcal{P}$, while $\Im$ corresponds to the subset of measures under which $Y_1, \ldots, Y_n$ are conditionally independent given X. The first assertion can be proven by definition. The second assertion is proven as follows. First, notice that $\mathbb{Q}(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n) = \mathcal{Q}(i_1, l, i_2, \ldots, i_n)$. Then, by summing over all i_j with $j = 2, \ldots, n$ we have $\mathbb{Q}(Y_1 = i_1, X = l) = \mathbf{Q}_1(i_1, l)$, since $\mathbf{Q}_j^T \mathbf{e} = \mathbf{e}$ for $j = 2, \ldots, n$. By summing over all i_j with $j = 1, \ldots, n$ we have $\mathbb{Q}(X = l) = \sum_{i_1=1}^{I_1} \mathbb{Q}_1(i_1, l)$ and thus $\mathbb{Q}(Y_1 = i_1 | X = l) = \frac{\mathbb{Q}(Y_1 = i_1, X = l)}{\mathbb{Q}(X = l)}$. Moreover, for $j = 2, \ldots, n$ and by summing over all i_m with $m \in \{1, \ldots, n\} - j$ then $\mathbb{Q}(Y_j = i_j | X = l) = \frac{\mathbb{Q}(Y_j = i_j, X = l)}{\mathbb{Q}(X = l)} = \frac{\sum_{i_1=1}^{I_1} \mathbf{Q}_1(i_1, l) \mathbf{Q}_j(i_j, l)}{\sum_{i_1=1}^{I_1} \mathbf{Q}_1(i_1, l)} = \mathbf{Q}_j(i_j, l)$. Finally, $\mathbb{Q}(Y_1 = i_1, \ldots, Y_n = i_n | X = l) = \frac{\mathbb{Q}(Y_1 = i_1, X = l, \ldots, Y_{n-1} = i_{n-1}, Y_n = i_n)}{\mathbb{Q}(X = l)} = \mathbb{Q}(Y_1 = i_1, X = l) \mathbb{Q}(Y_2 = i_2 | X = l) \ldots \mathbb{Q}(Y_n = i_n | X = l)$.

Lemma 2.1 *Tensor* $\mathcal{P}$ *admits exact factorization of* K *rank one tensors iff* $\wp \cap \Im \neq \emptyset$ *and an approximate NTF as* $\wp \cap \Im = \emptyset$.

A proof can be found in Appendix 15.

Now we generalize the probabilistic interpretation of NMF to an arbitrary order exact NTF problem. The probability tensor $\mathcal{P}$ admits an exact NTF (i.e., $\mathcal{P} = \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{q}_j^l$) iff there exist at least one measure $(\Omega, \mathcal{F})$ whose $Y = (Y_1, \ldots, Y_n)$ marginal is $\mathcal{P}$ and at the same time making $Y_1, \ldots, Y_n$ conditionally independent given X.

Proposition 2.2 *Given $\mathcal{P}$, function $(\mathcal{V}, \mathcal{Q}) \rightarrow D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q})$ attains a minimum on $\wp \times \Im$ and it is valid that*

$$\min_{\mathcal{Q} \in \aleph} D_{\mathrm{KL}}(\mathcal{P}||\mathcal{Q}) = \min_{\mathcal{V} \in \wp, \mathcal{Q} \in \Im} D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q}) \tag{55}$$

The proof can be found in Appendix 16.

Let $\mathcal{V}^{(\mathrm{opt})}$ and $\mathcal{Q}^{(\mathrm{opt})}$ be the tensors satisfying Proposition 2.2. If $\exists\ l_0$ such that $\sum_{i_1=1, i_2=1, \ldots, i_n=1}^{I_1, \ldots, I_n} \mathcal{V}^{(\mathrm{opt})}(i_1, l_0, i_2, \ldots, i_n) = 0$, then all $\mathcal{Q}^{(\mathrm{opt})}(i_1, l_0, i_2, \ldots, i_n) = 0$, as well. Equivalently, if $\exists\ l_0$, such that $\sum_{i_1=1, \ldots, i_n=1}^{I_1, \ldots, I_n} \mathcal{Q}^{(\mathrm{opt})}(i_1, l_0, i_2, \ldots, i_n) = 0$, then all $\mathcal{V}^{(\mathrm{opt})}(i_1, l_0, i_2, \ldots, i_n) = 0$, as well. In both cases the optimal factorization of $\mathcal{P}$ has rank less than K and in order to proceed to the factorization, we may omit all the columns from $\mathbf{Q}_1, \ldots, \mathbf{Q}_n$ that correspond to l_0.

5.2 Two partial subproblems

Now we shall try to solve the equivalent double minimization problem:

$$\min_{\mathcal{V} \in \wp, \mathcal{Q} \in \Im} D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q}). \tag{56}$$

We should solve two partial minimization problems. In the first problem, given $\mathcal{Q} \in \Im$, we minimize $D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q})$ over $\mathcal{V}$, while in the second one, given $\mathcal{V} \in \wp$ we minimize the divergence over $\mathcal{Q}$ given $\mathcal{V}$.

The unique solution $\mathcal{V}^{(\mathrm{opt})} = \mathcal{V}^{(\mathrm{opt})}(\mathcal{Q})$ can be easily computed analytically and is given by:

$$\mathcal{V}^{(\mathrm{opt})}(i_1, l, i_2, \ldots, i_n) = \frac{\mathcal{P}(i_1, \ldots, i_n)}{\sum_{l=1}^{K} \prod_{j=1}^{n} \mathbf{Q}_j(i_j, l)} \mathcal{Q}(i_1, l, i_2, \ldots, i_n) \tag{57}$$

where $\sum_{l=1}^{K} \prod_j \mathbf{Q}_j(i_j, l) = \sum_{l=1}^{K} \mathcal{Q}(i_1, l, i_2, \ldots, i_n)$.

For the second minimization problem, the unique solution is given by:

$$\mathbf{Q}_1^{(\mathrm{opt})} = \sum_{i_2=1, \ldots, i_n=1}^{I_2, \ldots, I_n} \mathcal{V}(i_1, l, i_2, \ldots, i_n) \tag{58}$$

and for $j = 2, \ldots, n$:

$$\mathbf{Q}_j^{(\text{opt})}(i_j, l) = \frac{\overline{\sum_{i_j}} \mathcal{V}(i_1, l, i_2, \ldots, i_n)}{\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathcal{V}(i_1, l, i_2, \ldots, i_n)} \tag{59}$$

These two minimization problems and their solutions have a direct probabilistic interpretation. This interpretation generalizes the discussion in Finesso and Spreij (2006) using arbitrary order tensors instead of order-2 tensors, i.e. matrices. In the first minimization given a distribution $\mathcal{Q}$, which makes the sequence of $Y = (Y_1, \ldots, Y_n)$ conditionally independent given X, the best solution represents the best set in $\wp$ with the marginal of Y being tensor $\mathcal{P}$. Let $\mathcal{V}^{(\text{opt})}$ be the optimal distribution $(Y_1, l, Y_2, \ldots, Y_n)$. Equation 57 can then be interpreted in terms of corresponding measures as:

$$\begin{aligned} &\mathbb{P}^{(\text{opt})}(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n) \\ &\quad = \mathbb{Q}(X = l | Y_1 = i_1, \ldots, Y_n = i_n) \mathcal{P}(i_1, i_2, \ldots, i_n) \end{aligned} \tag{60}$$

Notice that the conditional distributions of X given Y under $\mathbb{P}^{(\text{opt})}$ and $\mathbb{Q}$ are the same.

In the second optimization problem, one is given a distribution $\mathcal{Q}$, with the marginal of Y given by $\mathcal{P}$ and finds the best approximation to it in the set $\Im$ of distributions which make $Y = (Y_1, \ldots, Y_n)$ conditionally independent given X. Let $\mathcal{Q}^{(\text{opt})}$ denote the optimal distribution of $(Y_1, l, Y_2, \ldots, Y_{n-1}, Y_n)$. Equations 58 and 59 can be now be interpreted as:

$$\mathbb{Q}^{(\text{opt})}(Y_1 = i_1, X = l) = \mathbb{P}(Y_1 = i_1, X = l) \tag{61}$$

and for $j = 2, \ldots, n$

$$\mathbb{Q}^{(\text{opt})}(Y_j = i_j | X = l) = \mathbb{P}(Y_j = i_j | X = l), \tag{62}$$

respectively. That is, the optimal solution $\mathbb{Q}^{(\text{opt})}$ is such that the marginal distribution of (X, Y_1) under $\mathbb{P}$ and $\mathbb{Q}^{(\text{opt})}$ coincide, as well as, the marginal distributions of Y_j given X under $\mathbb{P}$ and $\mathbb{Q}^{(\text{opt})}$.

Lemma 2.2 *For fixed $\mathcal{Q}$ and $\mathcal{V}^{(\text{opt})} = \mathcal{V}^{(\text{opt})}(\mathcal{Q})$ it holds that, for any $\mathcal{V} \in \wp$*

$$D_{\text{KL}}(\mathcal{V} || \mathcal{Q}) = D_{\text{KL}}\left(\mathcal{V} || \mathcal{V}^{(\text{opt})}\right) + D_{\text{KL}}(\mathcal{V}^{(\text{opt})} || \mathcal{Q}) \tag{63}$$

moreover

$$D_{\text{KL}}\left(\mathcal{V}^{(\text{opt})} || \mathcal{Q}\right) = D_{\text{KL}}(\mathcal{P} || \mathcal{O}) \tag{64}$$

where

$$\mathcal{O}(i_1,\ldots,i_n)=\sum_{l=1}^{K}\mathcal{Q}(i_1,l,i_2,\ldots,l,i_n) \tag{65}$$

For fixed $\mathcal{V}$ *and* $\mathcal{Q}^{(\mathrm{opt})}=\mathcal{Q}^{(\mathrm{opt})}(\mathcal{V})$*, it holds that, for any* $\mathcal{Q}\in\Im$.

$$D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q})=D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q}^{(\mathrm{opt})})+D_{\mathrm{KL}}\left(\mathcal{Q}^{(\mathrm{opt})}||\mathcal{Q}\right). \tag{66}$$

The proof is provided in Appendix 17.

Before we produce the other two Pythagorean rules for NTF, we should define the following laws. Let the law of (U, V) under arbitrary probability measures $\mathbb{P}$ and $\mathbb{Q}$ by $\mathbb{P}^{U,V}$ and $\mathbb{Q}^{U,V}$. The conditional distributions of U given V are denoted by the matrices $\mathbb{P}^{U|V}$ and $\mathbb{Q}^{U|V}$, with the convention $\mathbb{P}^{U|V}(i,j)=\mathbb{P}(U=j|V=i)$ and likewise for $\mathbb{Q}^{U|V}$. Let us define the mean value operator $\mathbb{E}_{\mathcal{Z}}D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O})$

$$\mathbb{E}_{\mathcal{Z}}\quad D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O})=\sum\nolimits_{i_1,\ldots,i_n}\mathcal{Z}(i_1,\ldots,i_n)\mathcal{P}(i_1,\ldots,i_n)\ln\left(\frac{\mathcal{P}(i_1,\ldots,i_n)}{\mathcal{O}(i_1,\ldots,i_n)}\right). \tag{67}$$

Lemma 2.3 *It holds that*

$$D_{\mathrm{KL}}\left(\mathbb{P}^{U,V}||\mathbb{Q}^{U,V}\right)=\mathbb{E}_{\mathbb{P}}D_{\mathrm{KL}}\left(\mathbb{P}^{U|V}||\mathbb{Q}^{U|V}\right)+D_{\mathrm{KL}}\left(\mathbb{P}^{V}||\mathbb{Q}^{V}\right) \tag{68}$$

where

$$D_{\mathrm{KL}}\left(\mathbb{P}^{U|V}||\mathbb{Q}^{U|V}\right)=\sum_{j}\mathcal{P}(U=j|V)\log\frac{\mathcal{P}(U=j|V)}{\mathcal{O}(U=j|V)} \tag{69}$$

If moreover $V=(V_1,V_2)$ *and* U, V_2 *are conditionally independent given* V_1 *under* $\mathbb{Q}$*, then the first term in* (68) *can be written as:*

$$\mathbb{E}_{\mathbb{P}}D_{\mathrm{KL}}\left(\mathbb{P}^{U|V}||\mathbb{Q}^{U|V}\right)=\mathbb{E}_{\mathbb{P}}D_{\mathrm{KL}}\left(\mathbb{P}^{U|V}||\mathbb{P}^{U|V_1}\right)+\mathbb{E}_{\mathbb{P}}D_{\mathrm{KL}}\left(\mathbb{P}^{U|V_1}||\mathbb{Q}^{U|V_1}\right) \tag{70}$$

Now we will reinterpret the Pythagorean rules (63), (64) and (66) using probabilistic terms. In that way we generalize the probabilistic interpretation of NMF pythagorean rules in Finesso and Spreij (2006). The first optimization problem of (56) can be reinterpreted in probabilistic terms using similar lines as Finesso and Spreij (2006). That is, given $\mathcal{Q}$, we are to find its best approximation within $\wp$. Let $\mathbb{Q}$ correspond to a given $\mathcal{Q}$ and the $\mathbb{P}$ corresponds to a generic $\mathcal{V}\in\wp$. Choosing now, $U=X$ and $V=Y=(Y_1,\ldots,Y_n)$ in Lemma 2.3, Eq. 68 reads

$$D_{\mathrm{KL}}\left(\mathcal{V}||\mathcal{Q}\right)=\mathbb{E}_{\mathbb{P}}D_{\mathrm{KL}}\left(\mathbb{P}^{X|Y}||\mathbb{Q}^{X|Y}\right)+D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{Q}\right). \tag{71}$$

Moreover, (71) is equivalent to (63). That is, the optimization problem is equivalent to the minimization of $\mathbb{E}_{\mathbb{P}}\left(\mathbb{P}^{X|Y}||\mathbb{Q}^{X|Y}\right)$ with respect to $\mathcal{V} \in \wp$, which is attained (with value 0) at $\mathbb{P}^{(\text{opt})}$ with $\mathbb{P}^{X|Y}_{(\text{opt})} = \mathbb{Q}^{X|Y}$ and $\mathbb{P}^{Y}_{(\text{opt})} = \mathcal{P}$.

That is, for the second optimization problem (i.e, minimizing $D_{\text{KL}}(\mathcal{V}||\mathcal{Q})$ given $\mathcal{V}$) we have the following. Given $\mathcal{V}$ we are to find its best approximation within $\mathcal{Q}$. Let $\mathbb{P}$ correspond to given $\mathcal{V}$ and $\mathbb{Q}$ correspond to a generic $\mathcal{Q} \in \Im$. Choosing $U = \{Y_2, \ldots, Y_n\}$, $V_1 = X$ and $V_2 = Y_1$ in Lemma 2.3, and remembering that under any $\mathcal{Q} \in \Im$ the random variables $Y_1, Y_2, \ldots, Y_n$ are conditionally independent given X, Eq. 68 combined with (70) now reads:

$$\begin{aligned} &D_{\text{KL}}(\mathcal{V}||\mathcal{Q}) \\ &= \mathbb{E}_{\mathbb{P}} D_{\text{KL}}\left(\mathbb{P}^{Y_2,\ldots,Y_n|X,Y_1}||\mathbb{P}^{Y_1,\ldots,Y_n|X}\right) + \mathbb{E}_{\mathbb{P}} D_{\text{KL}}\left(\mathbb{P}^{Y_2,\ldots,Y_n|X}||\mathbb{Q}^{Y_2,\ldots,Y_n|X}\right) \\ &+ D_{\text{KL}}\left(\mathbb{P}^{Y_1,X}||\mathbb{Q}^{Y_1,X}\right) = \sum_{j=2}^{n} \mathbb{E}_{\mathbb{P}} D_{\text{KL}}\left(\mathbb{P}^{Y_j|X}||\mathbb{Q}^{Y_j|X}\right) + D_{\text{KL}}\left(\mathbb{P}^{Y_1,X}||\mathbb{Q}^{Y_1,X}\right) \end{aligned} \tag{72}$$

as shown analytically in Appendix 18. In the special case of NMF (Finesso and Spreij 2006) (i.e., only two random variables Y_1 and Y_2) the above Pythagorean rule is written as $D_{\text{KL}}(\mathcal{V}||\mathcal{Q}) = \mathbb{E}_{\mathbb{P}} D_{\text{KL}}\left(\mathbb{P}^{Y_2|X}||\mathbb{Q}^{Y_2|X}\right) + D_{\text{KL}}\left(\mathbb{P}^{Y_1,X}||\mathbb{Q}^{Y_1,X}\right)$.

The problem is equivalent to the minimization of $\sum_{j=2}^{n} \mathbb{E}_{\mathbb{P}} D_{\text{KL}}\left(\mathbb{P}^{Y_j|X}||\mathbb{Q}^{Y_j|X}\right)$ and $D_{\text{KL}}\left(\mathbb{P}^{Y_1,X}||\mathbb{Q}^{Y_1,X}\right)$ with respect to $\mathcal{Q} \in \Im$. Both these minima are attained (both with value 0) at $\mathbb{Q}^{(\text{opt})}$ with $\mathbb{Q}^{Y_j|X}_{(\text{opt})} = \mathbb{P}^{Y_j|X}$ for $j = 2, \ldots, n$ and $\mathbb{Q}^{Y_1,X}_{(\text{opt})} = \mathbb{P}^{Y_1,X}$. Note that X have the same distribution under $\mathbb{P}$ and $\mathbb{Q}^{(\text{opt})}$. To derive probabilistically the corresponding generalized Pythagorean rule we notice that:

$$\begin{aligned} D_{\text{KL}}(\mathcal{V}||\mathcal{Q}) - D_{\text{KL}}(\mathcal{V}||\mathcal{Q}^{(\text{opt})}) = &\sum_{j=2}^{n} \mathbb{E}_{\mathbb{P}} D_{\text{KL}}\left(\mathbb{Q}^{Y_j|X}_{(\text{opt})}||\mathbb{Q}^{Y_j|X}\right) \\ &+ D_{\text{KL}}\left(\mathbb{Q}^{Y_1,X}_{(\text{opt})}||\mathbb{Q}^{Y_1,X}\right). \end{aligned} \tag{73}$$

On the right hand side of (73) we can, by conditional independence, replace $\mathbb{E}_{\mathbb{Q}^{(\text{opt})}} D_{\text{KL}}(\mathbb{Q}^{Y_j|X}_{(\text{opt})}||\mathbb{Q}^{Y_j|X})$, for $j = 2, \ldots, n$ with $\mathbb{E}_{\mathbb{Q}^{(\text{opt})}} D_{\text{KL}}(\mathbb{Q}^{Y_j|X,Y_1}_{(\text{opt})}||\mathbb{Q}^{Y_1|X,Y_1})$. By another application of (68), we see that $D_{\text{KL}}(\mathcal{V}||\mathcal{Q}) - D_{\text{KL}}\left(\mathcal{V}||\mathcal{Q}^{(\text{opt})}\right) = D_{\text{KL}}\left(\mathcal{Q}^{(\text{opt})}||\mathcal{Q}\right)$, which is the second Pythagorean rule (66).

5.2.1 Alternating minimization procedure

As in Finesso and Spreij (2006), we set up the alternating minimization algorithm for obtaining $\min_{\mathcal{O}} D_{\text{KL}}(\mathcal{P}||\mathcal{O})$, where $\mathcal{P}$ is a given nonnegative tensor of arbitrary order. In view of Proposition 2.2. we can lift this problem in $\wp \times \Im$ space. Starting with an arbitrary $\mathcal{Q}^{(0)} \in \Im$ with positive elements, we adopt the following minimization

scheme

$$\rightarrow \mathcal{Q}^{(t)} \rightarrow \mathcal{V}^{(t)} \rightarrow \mathcal{Q}^{(t+1)} \rightarrow \mathcal{V}^{(t+1)} \rightarrow \tag{74}$$

where $\mathcal{V}^{(t)} = \mathcal{V}^{(\text{opt})}(\mathcal{Q}^{(t)})$, $\mathcal{Q}^{(t+1)} = \mathcal{Q}^{(\text{opt})}(\mathcal{V}^{(t)})$. As we can see by (74) the update gain is independent of the order of the tensors.

To relate this algorithm to the one defined by formulae (48) and (49), we combine two steps of the alternative minimization at a time. From the algorithms (74) we get:

$$\mathcal{Q}^{(t+1)} = \mathcal{Q}^{(\text{opt})}\left(\mathcal{V}^{(\text{opt})}(\mathcal{Q}^{(t)})\right). \tag{75}$$

Computing the optimal solutions according to (57), (58) and (59) one gets from there the update rules (48) and (49). The Pythagorean rules allow us to compute easily the gain $D_{\text{KL}}(\mathcal{P}||\mathcal{O}^{(t)}) - D_{\text{KL}}(\mathcal{P}||\mathcal{O}^{(t+1)})$ of the algorithm.

Proposition 2.3 *The update gain at each iteration of the algorithm (75) in terms of the tensors $\mathcal{O}^{(t)}$ is given by:*

$$D_{\text{KL}}\left(\mathcal{P}||\mathcal{O}^{(t)}\right) - D_{\text{KL}}\left(\mathcal{P}||\mathcal{O}^{(t+1)}\right) = D_{\text{KL}}\left(\mathcal{V}^{(t)}||\mathcal{V}^{(t+1)}\right) + D_{\text{KL}}\left(\mathcal{Q}^{(t+1)}||\mathcal{Q}^{(t)}\right) \tag{76}$$

The proof is provided in Appendix 19.

If for matrices $\left(\mathbf{Q}_1^{(0)}, \ldots, \mathbf{Q}_n^{(0)}\right)$ we have that ${q^l_{i_j j}}^{(0)} > 0$ then under the iterations in (48) and in (49) we shall have ${q^l_{i_j j}}^{(t)} > 0$ with $t > 0$. If in the tth step the update gain is zero then we will have $D_{\text{KL}}(\mathcal{Q}^{(t)}||\mathcal{Q}^{(t-1)}) = 0$. Hence, tensors $\mathcal{Q}^{(t)}$ and $\mathcal{Q}^{(t-1)}$ are identical. From this it follows by summation that $\mathbf{Q}_1^{(t)} = \mathbf{Q}_1^{(t-1)}$. Moreover, we shall have that $\mathbf{Q}_1^{(t+1)}(i_1, l) \ldots \mathbf{Q}_n^{(t+1)}(i_n, l) = \mathbf{Q}_1^{(t)}(i_1, l) \ldots \mathbf{Q}_n^{(t)}(i_n, l)$, since $\mathbf{Q}_1^{(t)} = \mathbf{Q}_1^{(t-1)}$, $\mathbf{Q}_1^{(t-1)}$ is strictly positive and by summing for all $a \in \{2, \ldots, j-1, j+1, \ldots, n\}$ we have that $\mathbf{Q}_j^{(t)} = \mathbf{Q}_j^{(t-1)}$ for all $j \in \{2, \ldots, n\}$. Hence, the update rules strictly decrease the objective function until the algorithm reaches a fixed point.

5.3 Auxiliary functions

We consider now the minimization of $D_{\text{KL}}(\mathcal{P}||\mathcal{O})$ and its lifted version i.e., the minimization of $D_{\text{KL}}(\mathcal{V}||\mathcal{Q})$. We shall consider that the problem starts by noting that $\mathcal{Q}^{(t+1)}$ is found by minimizing $\mathcal{Q}' \rightarrow D_{\text{KL}}(\mathcal{V}^{(\text{opt})}(\mathcal{O}^{(t)})||\mathcal{O}')$. This strongly motivates the choice of function

$$\left(\mathcal{Q}, \mathcal{Q}'\right) \rightarrow W(\mathcal{Q}, \mathcal{Q}') = D_{\text{KL}}\left(\mathcal{V}^{\text{opt}}(\mathcal{Q}^{(t)})||\mathcal{Q}'\right) \tag{77}$$

as an auxiliary function for minimizing $D_{\text{KL}}(\mathcal{P}||\mathcal{O})$ with respect to $\mathcal{O}$.

Using the decomposition of the divergence in Eq. 78 we can rewrite W as

$$W\left(\mathcal{Q},\mathcal{Q}'\right)=D_{\mathrm{KL}}\left(\mathcal{V}^{\mathrm{opt}}(\mathcal{Q}^{(t)})||\mathcal{Q}'\right)=D_{\mathrm{KL}}\left(\mathbb{P}^{Y}_{(\mathrm{opt})}||\mathbb{Q}'^{Y}\right)+\mathbb{E}_{\mathbb{P}^{(\mathrm{opt})}}D_{\mathrm{KL}}\left(\mathbb{P}^{X|Y}_{(\mathrm{opt})}||\mathbb{Q}'^{X|Y}\right) \tag{78}$$

Since $\mathbb{P}^{X|Y}_{(\mathrm{opt})}=\mathbb{Q}^{X|Y}$, and $\mathbb{P}^{Y}_{(\mathrm{opt})}=\mathcal{P}$ we write (78) as

$$W\left(\mathcal{Q},\mathcal{Q}'\right)=D_{\mathrm{KL}}\left(\mathcal{P}||\mathbb{Q}'^{Y}\right)+\mathbb{E}_{\mathcal{P}}D_{\mathrm{KL}}\left(\mathbb{Q}^{X|Y}||\mathbb{Q}'^{X|Y}\right) \tag{79}$$

From (79) it follows that $W\left(\mathcal{Q},\mathcal{Q}'\right)\geq D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}'\right)$, and that $W\left(\mathcal{Q},\mathcal{Q}\right)=D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}\right)$, which are the properties of an auxiliary function for $D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}\right)$.

For our problem one can find n auxiliary functions for the original minimization of $D_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l\right)$ (the same auxiliary functions given in (33)). In every auxiliary function for $\mathbf{Q}_j$, $j=1,\ldots,n$ we fix all $m\in\{1,\ldots,n\}-\{j\}$. As in Finesso and Spreij (2006) the auxiliary function $W\left(\mathcal{Q},\mathcal{Q}'\right)$ can be denoted as $W\left(\mathbf{Q}_1,\ldots,\mathbf{Q}_n,\mathbf{Q}'_1,\ldots,\mathbf{Q}'_n\right)$. The auxiliary function minimization with fixed $\mathbf{Q}_m$ with $m\in\{1,\ldots,n\}-\{j\}$ can be taken as

$$\mathbf{Q}'_j\rightarrow W^j_{\mathcal{O}}\left(\mathbf{Q}'_j\right)=W\left(\mathbf{Q}_1,\ldots,\mathbf{Q}_n,\mathbf{Q}_1,\ldots,\mathbf{Q}_{j-1},\mathbf{Q}'_j,\mathbf{Q}_{j+1},\ldots,\mathbf{Q}_n\right) \tag{80}$$

Now, we shall generalize the update gains in Finesso and Spreij (2006) and provide the update gains for arbitrary order NTF decompositions.

Lemma 2.4 *Consider the auxiliary functions $W\left(\mathcal{Q},\mathcal{Q}'\right)$ and n auxiliary functions $W\left(\mathbf{Q}_1,\ldots,\mathbf{Q}_n,\mathbf{Q}_1,\ldots,\mathbf{Q}_{j-1},\mathbf{Q}'_j,\mathbf{Q}_{j+1},\ldots,\mathbf{Q}_n\right)$. Denote by $\mathbf{Q}'_j$ the minimizers of auxiliary functions in all $n+1$ cases. Then, the following Lemma holds.*

$$D_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l\right)-W_{\mathcal{Q}}\left(\mathbf{Q}'_1\right)=D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_1,X}||\mathbb{Q}^{Y_1,X}\right) \tag{81}$$

$$D_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l\right)-W_{\mathcal{Q}}\left(\mathbf{Q}'_j\right)=\mathbb{E}_{\mathbb{P}^{(\mathrm{opt})}}D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_j|X}||\mathbb{Q}^{Y_j|X}\right) \tag{82}$$

$$D_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l\right)-W\left(\mathbf{Q}_1,\ldots,\mathbf{Q}_n,\mathbf{Q}'_1,\ldots,\mathbf{Q}'_n\right)=D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_1,X}||\mathbb{Q}^{Y_1,X}\right)+\sum_{j=2}^{n}\mathbb{E}_{\mathbb{Q}'}D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_j|X}||\mathbb{Q}^{Y_j|X}\right) \tag{83}$$

The proof of the above can be found in Appendix 20.

Corollary 2.2 *The update gain of the multiplicative update rules in (48) and (49) is given by:*

$$D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}^{(t)}\right) - D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}^{(t+1)}\right) = D_{\mathrm{KL}}\left(\mathbb{Q}^{(t+1)^{Y_1,X}}||\mathbb{Q}^{(t)^{Y_1,X}}\right) + \sum_{j=2}^{n} \mathbb{E}_{\mathbb{Q}^{(t+1)}} D_{\mathrm{KL}}\left(\mathbb{Q}^{(t+1)^{Y_j|X}}||\mathbb{Q}^{(t)^{Y_j|X}}\right) + \mathbb{E}_{\mathbb{P}} D_{\mathrm{KL}}\left(\mathbb{Q}^{(t)^{X|Y}}||\mathbb{Q}^{(t+1)^{X|Y}}\right). \tag{84}$$

In order to prove the above we start by writing:

$$D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}^{(t)}\right) - D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}^{(t+1)}\right) = D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}^{(t)}\right) - W\left(\mathcal{O}^{(t)}, \mathcal{O}^{(t+1)}\right) + W\left(\mathcal{O}^{(t)}, \mathcal{O}^{(t+1)}\right) - D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{O}^{(t+1)}\right) \tag{85}$$

and then, using (79) and (83) the corollary is proven. Finally, by using (76) we can prove that:

$$\begin{aligned} D_{\mathrm{KL}}\left(\mathcal{V}^{(t)}||\mathcal{V}^{(t+1)}\right) &= \mathbb{E}_{\mathcal{P}}\left(\mathbb{Q}^{(t)^{X|Y}}||\mathbb{Q}^{(t+1)^{X|Y}}\right) \\ D_{\mathrm{KL}}\left(\mathcal{Q}^{(t)}||\mathcal{Q}^{(t+1)}\right) &= D_{\mathrm{KL}}\left(\mathbb{Q}^{(t)^{X|Y}}||\mathbb{Q}^{(t+1)^{X|Y}}\right) \\ &+ \sum_{j=2}^{n} E_{\mathbb{Q}^{(t+1)}} D_{\mathrm{KL}}\left(\mathbb{Q}^{(t+1)^{Y_j|X}}||\mathbb{Q}^{(t)^{Y_j|X}}\right) \end{aligned} \tag{86}$$

6 Probabilistic latent tensor variable analysis

In the previous section we studied some of the properties of the NTF optimization problem in (43) and the corresponding algorithm based on multiplicative update rules. In this section we shall try to relate the above algorithm with probabilistic tensor latent component analysis models. In Shashanka et al. (2008) the authors commented about the possible extension of PLSA and PLVA models (the algorithms of which are implemented using update rules similar to NMF) using arbitrary order tensors but no algorithm was proposed. Here we shall show that the algorithm for solving (43) can indeed be used for multilinear PLTVA and by introducing additional factors we propose algorithms for PLTVA which are the generalization of the NMF algorithms used in Shashanka et al. (2008) and Gaussier and Goutte (2005). In the following, we generalize both the symmetric and asymmetric PLSA models.

6.1 The symmetric model

In the symmetric PLSA model for the two random variable case we seek for the latent probabilities matrices for $P(x_2|z)$ and $P(x_1, z)$ (or $P(x_1|z)$). Let $x_1, \ldots, x_n$ be random

variables. Then approximation (16) can be generalized as:

$$P(x_1, \ldots, x_n) \approx \sum_{z, z=1,\ldots,K} P(x_1, z) \prod_{j=2}^{n} P(x_j|z) \tag{87}$$

In an EM manner the above is solved as:

$$P^{(t)}(z|x_1, \ldots, x_n) = \frac{P^{(t-1)}(x_1, z) \prod_{j=2}^{n} P^{(t-1)}(x_j|z)}{\sum_{z\in\{1,2,\ldots,K\}} P^{(t-1)}(x_1, z) \prod_{j=2}^{n} P^{(t-1)}(x_j|z)} \tag{88}$$

and

$$\begin{aligned} P^{(t)}(x_1, z) &= \sum_{x_2,\ldots,x_n} P(x_1, \ldots, x_n) P^{(t)}(z|x_1, \ldots, x_n) \\ P^{(t)}(x_j|z) &= \frac{\sum_{x_1,\ldots,x_{j-1},x_{j+1},\ldots,x_n} P(x_1, \ldots, x_n) P^{(t)}(z|x_1, \ldots, x_n)}{\sum_{x_1,\ldots,x_n} P(x_1, \ldots, x_n) P^{(t)}(z|x_1, \ldots, x_n)}, \quad j = 2, \ldots, n. \end{aligned} \tag{89}$$

We can easily verify that the above update rules (88) and (89) are equivalent to update rules (57), (58) and (59) and also to update rules (48) and (49). Thus, approximation (87) is equivalent to optimization problem (43).

Now, we shall expand further the approximation (87) by allowing another factor $P(z)$, which is naturally introduced by expanding $P(x_1, z) = P(z)P(x_1|z)$. Thus, factorization (87) becomes:

$$P(x_1, \ldots, x_n) \approx \sum_{z, z=1,\ldots,K} P(z) \prod_{j=1}^{n} P(x_j|z). \tag{90}$$

In an EM approach the above problem is solved as:

$$P^{(t)}(z|x_1, \ldots, x_n) = \frac{P^{(t-1)}(z) \prod_{j=1}^{n} P^{(t-1)}(x_j|z)}{\sum_{z\in\{1,2,\ldots,K\}} P^{(t-1)}(z) \prod_{j=1}^{n} P^{(t-1)}(x_j|z)} \tag{91}$$

and

$$\begin{aligned} P^{(t)}(x_j|z) &= \frac{\overline{\sum_{x_j}} P(x_1, \ldots, x_n) P^{(t)}(z|x_1, \ldots, x_n)}{\sum_{x_1,\ldots,x_n} P(x_1, \ldots, x_n) P^{(t)}(z|x_1, \ldots, x_n)} \\ P^{(t)}(z) &= \sum_{x_1,\ldots,x_n} P(x_1, \ldots, x_n) P^{(t)}(z|x_1, \ldots, x_n). \end{aligned} \tag{92}$$

By substituting (91) into (92) we get:

$$P^{(t)}(x_j|z) = \frac{\overline{\sum_{x_j}} P(x_1,\ldots,x_n) \frac{P^{(t-1)}(z)\prod_{\substack{r=1\\(j,t)}}^{n} P(x_r|z)}{\sum_{z\in\{1,2,\ldots,K\}} P^{(t-1)}(z)\prod_{\substack{r=1\\(j,t)}}^{n} P(x_r|z)}}{\sum_{x_1,\ldots,x_n} P(x_1,\ldots,x_n) \frac{P^{(t-1)}(z)\prod_{\substack{r=1\\(j,t)}}^{n} P(x_r|z)}{\sum_{z\in\{1,2,\ldots,K\}} P^{(t-1)}(z)\prod_{\substack{r=1\\(j,t)}}^{n} P(x_r|z)}}$$

$$P^{(t)}(z) = \sum_{x_1,\ldots,x_n} P(x_1,\ldots,x_n) \frac{P^{(t-1)}(z)\prod_{j=1}^{n} P^{(t)}(x_j|z)}{\sum_{z\in\{1,2,\ldots,K\}} P^{(t-1)}(z)\prod_{j=1}^{n} P^{(t)}(x_j|z)}. \tag{93}$$

Now, let $\mathcal{P}$ be the probability tensor with $p_{i_1\ldots i_n} = \mathbb{P}(x_1 = i_1, x_2 = i_2, \ldots, x_n = i_n)$ and let matrices $[\mathbf{Q}_j]_{i_j l} = \mathbb{P}(x_j = i_j|z = l)$ and vector $\mathbf{a}$ with $[\mathbf{a}]_l = \mathbb{P}(z = l)$. Then, approximation (90) is written as:

$$\mathcal{P} \approx \mathcal{R} = \sum_{l=1}^{K} a_l \bigcirc_{j=1}^{n} \mathbf{q}_j^l. \tag{94}$$

In a matricized form the above nonnegative approximation can be written as:

$$\mathbf{P}_{(j)} \approx \mathbf{R}_{(j)} = \mathbf{Q}_j \mathbf{Z}_{(j)}^T = \mathbf{Q}_j \mathbf{A} \left(\overline{\bigodot_{j}} \mathbf{Q}_i \right)^T \tag{95}$$

for $j = 1, \ldots, n$ where $\mathbf{A} = \mathbf{diag}(\mathbf{a})$ and $\mathbf{Z}_{(j)} = \left(\overline{\bigodot_{j}} \mathbf{Q}_i \right) \mathbf{A}$. In terms of $\mathbf{A}$ it can be written as:

$$\mathrm{vec}(\mathbf{P}_{(1)}) \approx \left(\mathbf{Q}_1 \odot \left(\overline{\bigodot_{1}} \mathbf{Q}_i \right) \right) \mathbf{a}. \tag{96}$$

Using tensor unfolding and matricization procedures, the update rules (93) can be written as:

$$\begin{aligned} \tilde{\mathbf{Q}}_j^{(t)} &= \mathbf{Q}_j{}^{(t-1)} \bullet \frac{\mathbf{P}_{(j)}}{\mathbf{R}_{(j)}^{(t-1)}} \mathbf{Z}_{(j)}^{(t-1)\,T} \\ \mathbf{Q}_j{}^{(t)} &= \frac{\tilde{\mathbf{Q}}_j^{(t)}}{\mathbf{E}\tilde{\mathbf{Q}}_j^{(t)}} \end{aligned} \tag{97}$$

where $\mathbf{R}_{(j)}^{(t-1)} = \mathbf{Q}_j^{(t-1)} \mathbf{Z}_{(j)}{}^{(t-1)\,T}$ and $\mathbf{Z}_{(j)}^{(t-1)} = \mathbf{Q}_n^{(t-1)} \odot \ldots \odot \mathbf{Q}_{j+1}^{(t-1)} \odot \mathbf{Q}_{j-1}^{(t)} \odot \ldots \odot \mathbf{Q}_1^{(t)}$.

The update rule for the probabilities of the latent variables is:

$$\mathbf{a}^{(t)} = \mathbf{a}^{(t-1)} \bullet \left(\left(\mathbf{Q}_1^{(t)} \odot \left(\overline{\bigodot_1} \mathbf{Q}_i^{(t)} \right) \right)^T \frac{\text{vec}(\mathbf{P}_{(1)})}{\left(\mathbf{Q}_1^{(t)} \odot \left(\overline{\bigodot_1} \mathbf{Q}_i^{(t)} \right) \right) \mathbf{a}^{(t-1)}} \right). \quad (98)$$

We can easily verify that the above update rules are the solution of the following optimization problem:

$$\min_{\mathbf{Q}_1,\ldots,\mathbf{Q}_n : \mathbf{Q}_j{}^T \mathbf{e}=\mathbf{e}, \mathbf{e}^T \mathbf{a}=1} D_{\text{KL}} \left(\mathcal{P} || \sum_{l=1}^{K} a_l \bigcirc_{j=1}^{n} \mathbf{q}_j^l \right). \quad (99)$$

In order to use the above algorithm for the factorization of an arbitrary tensor $\mathcal{X} \approx \sum_{l=1}^{K} a_l \bigcirc_j \mathbf{u}_j^l$ with $w_1 = \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathcal{X}(i_1, \ldots, i_n)$ we should normalize the tensor into $\mathcal{P} = \frac{1}{w_1} \mathcal{X}$, then use update rules (97). Finally, the factors of the decomposition are given by $\mathbf{b}^{(\text{opt})} = w_1 \mathbf{a}^{(\text{opt})}$ and $\mathbf{U}_j^{(\text{opt})} = \mathbf{Q}_j^{(\text{opt})}$.

6.2 Asymmetric model

Now we generalize the asymmetric PLSA model for the arbitrary order tensor case. In the asymmetric PLSA model for the two random variable case we seek for the latent probabilities matrices for $P(x_1|z)$ and $P(z|x_2)$. An example of the asymmetric model decomposition is the application of NMF to facial images. In this case the images are vectorized. The vectors are then normalized in order to sum up to one (i.e., in order to form probability mass functions). Then, matrix $\mathbf{P} = [P(x_1 = i|x_2 = j)]$ contains as columns the facial images (x_1 is the random variable that corresponds to feature dimensionality and x_2 is the random variable that corresponds to number of faces in the dataset). The decomposition (16) can be reformulated as:

$$P(x_1|x_2) = \sum_{z \in \{1,\ldots,K\}} P(z|x_2) P(x_1|z) \quad (100)$$

then using the EM in (18) can be reformulated as:

$$P^{(t)}(z|x_1, x_2) = \frac{P^{(t-1)}(z|x_2) P^{(t-1)}(x_1|z)}{\sum_{z \in \{1,\ldots,K\}} P^{(t-1)}(x_1|z) P^{(t-1)}(z|x_2)} \quad (101)$$

and

$$P^{(t)}(z|x_2) = \frac{\sum_{x_1} P(x_1, x_2, z)}{\sum_{x_1, z\in\{1,\ldots,K\}} P(x_1, x_2, z)} = \frac{\sum_{x_1} P(x_1|z)P(z|x_1, x_2)}{\sum_{x_1, z\in\{1,\ldots,K\}} P(x_1|z)P(z|x_1, x_2)}$$
$$= P^{(t-1)}(z|x_2) \frac{\sum_{x_1} \frac{P(x_1|x_2)}{\sum_{z\in\{1,\ldots,K\}} P^{(t-1)}(x_1|z)P^{(t-1)}(z|x_2)} P^{(t-1)}(x_1|z)}{\sum_{x_1, z\in\{1,\ldots,K\}} \frac{P(x_1|x_2)}{\sum_{z\in\{1,\ldots,K\}} P^{(t-1)}(x_1|z)P^{(t-1)}(z|x_2)} P^{(t-1)}(z|x_2)P^{(t-1)}(x_1|z)} \tag{102}$$

and

$$P^{(t)}(x_1|z) = \frac{\sum_{x_2} P(x_1, x_2, z)}{\sum_{x_1,x_2} P(x_1, x_2, z)} = \frac{\sum_{x_2} P(x_2)P(x_1|x_2)P(z|x_1, x_2)}{\sum_{x_1,x_2} P(x_2)P(x_1|x_2)P(z|x_1, x_2)}$$
$$= P^{(t-1)}(x_1|z) \frac{\sum_{x_2} \frac{P(x_1|x_2)}{\sum_{z\in\{1,\ldots,K\}} P^{(t-1)}(x_1|z)P^{(t)}(z|x_2)} P(x_2)P^{(t)}(z|x_2)}{\sum_{x_1,x_2} \frac{P(x_1|x_2)}{\sum_{z\in\{1,\ldots,K\}} P^{(t-1)}(x_1|z)P^{(t)}(z|x_2)} P(x_2)P^{(t)}(z|x_2)P^{(t-1)}(x_1|z)}. \tag{103}$$

Let that the images are not vectorized, then each of the facial image is represented by a matrix and all the faces together for a tensor $\mathcal{P} = [P(x_1 = i_1, x_2 = i_2|x_3 = i_3)]$ where x_1 and x_2 are the random variables that correspond to height and width of the facial images and x_3 is the random variable that corresponds to the number of facial images in the dataset. In the general n-order case we have:

$$P(x_1, \ldots, x_n|x_{n+1}) = \sum_{z\in\{1,\ldots,K\}} P(z|x_{n+1})P(x_1, \ldots, x_n|z)$$
$$= \sum_{z\in\{1,\ldots,K\}} P(z|x_{n+1}) \prod_{j=1}^{n} P(x_j|z) \tag{104}$$

the above model in an EM manner can be solved as:

$$P^{(t)}(z|x_1, \ldots, x_{n+1}) = \frac{P^{(t-1)}(z|x_{n+1}) \prod_{j=1}^{n} P^{(t-1)}(x_j|z)}{\sum_{z\in\{1,\ldots,K\}} P^{(t-1)}(z|x_{n+1}) \prod_{j=1}^{n} P^{(t-1)}(x_j|z)} \tag{105}$$

and

$$P^{(t)}(z|x_{n+1}) = P^{(t-1)}(z|x_{n+1}) \frac{\overline{\sum_{x_{n+1}}} \frac{P(x_1,\ldots,x_n|x_{n+1})}{\sum_{z\in\{1,\ldots,K\}} P^{(t-1)}(z|x_{n+1}) \prod_{r=1}^{n} P^{(t-1)}(x_r|z)} \prod_{r=1}^{n} P^{(t-1)}(x_r|z)}{\overline{\sum_{x_j}} \sum_{z\in\{1,\ldots,K\}} \frac{P(x_1,\ldots,x_n|x_{n+1})}{\sum_{z\in\{1,\ldots,K\}} P^{(t-1)}(z|x_{n+1}) \prod_{r=1}^{n} P^{(t-1)}(x_r|z)} P^{(t-1)}(z|x_{n+1}) \prod_{j=1}^{n} P^{(t-1)}(x_j|z)} \tag{106}$$

and

$$P^{(t)}(x_j|z) = P^{(t-1)}(x_j|z) \frac{\overline{\sum_{x_j} \frac{P(x_1,\ldots,x_n|x_{n+1})}{\sum_{z\in\{1,\ldots,K\}} P^{(t)}(z|x_{n+1}) \prod_{\substack{r=1\\(j,t)}}^{n} P(x_r|z)} P(x_{n+1}) P^{(t)}(z|x_{n+1}) \prod_{\substack{r=1,r\neq j\\(j,t)}}^{n} P(x_r|z)}}{\sum_{x_1,\ldots,x_{n+1}} \frac{P(x_1,\ldots,x_n|x_{n+1})}{\sum_{z\in\{1,\ldots,K\}} P^{(t)}(z|x_{n+1}) \prod_{\substack{r=1\\(j,t)}}^{n} P(x_r|z)} P(x_{n+1}) P^{(t)}(z|x_{n+1}) \prod_{\substack{r=1\\(j,t)}}^{n} P(x_j|z)} \tag{107}$$

Then, by defining tensor $\mathcal{P}$ with elements $\mathcal{P}(i_1,\ldots,i_n) = \mathbb{P}(x_1 = i_1,\ldots,x_n = i_n|x_{n+1} = i_{n+1})$ and $\mathbf{Q}_j(i_j,l) = \mathbb{P}(x_1 = i_1|z = l)$ for $j = 1,\ldots,n$ and $\mathbf{Q}_{n+1}(i_{n+1},l) = \mathbb{P}(z = l|x_{n+1} = i_{n+1})$. Then, we formulate the following optimization problem:

$$\min_{\mathbf{Q}_1,\ldots,\mathbf{Q}_{n+1}:\mathbf{Q}_j{}^T\mathbf{e}=\mathbf{e},\mathbf{Q}_{n+1}\mathbf{e}=\mathbf{e}} D_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n+1}\mathbf{q}_j^l\right). \tag{108}$$

Using tensor unfolding and matricization procedures the update rules (93) can be written as:

$$\begin{aligned}\tilde{\mathbf{Q}}_j^{(t)} &= \mathbf{Q}_j{}^{(t-1)} \bullet \frac{\mathbf{P}_{(j)}}{\mathbf{R}_{(j)}^{(t-1)}}[\bar{\mathbf{Z}}_{(j)}^{(t-1)}]^T \\ \mathbf{Q}_j{}^{(t)} &= \frac{\tilde{\mathbf{Q}}_j^{(t)}}{\mathbf{E}\tilde{\mathbf{Q}}_j^{(t)}}\end{aligned} \tag{109}$$

for $j = 1,\ldots,n$ where $\mathbf{R}_{(j)}^{(t-1)} = \mathbf{Q}_j^{(t-1)}\mathbf{Z}_{(j)}{}^{(t-1)^T}$, $\mathbf{Z}_{(j)}{}^{(t-1)} = \mathbf{Q}_{n+1}^{(t-1)} \odot \mathbf{Q}_n^{(t-1)} \odot \ldots \odot \mathbf{Q}_{j+1}^{(t-1)} \odot \mathbf{Q}_{j-1}^{(t)} \odot \ldots \odot \mathbf{Q}_1^{(t)}$ and $\bar{\mathbf{Z}}_{(j)}^{(t-1)} = \bar{\mathbf{Q}}_{n+1}^{(t-1)} \odot \mathbf{Q}_n^{(t-1)} \odot \ldots \odot \mathbf{Q}_{j+1}^{(t-1)} \odot \mathbf{Q}_{j-1}^{(t)} \odot \ldots \odot \mathbf{Q}_1^{(t)}$, where $\bar{\mathbf{Q}}_{n+1}^{(t)} = \mathbf{B}\mathbf{Q}_{n+1}^{(t)}$ and $\mathbf{B} = \mathbf{diag}(\mathbf{b})$ is a $I_{n+1} \times I_{n+1}$ diagonal matrix with $\mathbf{b}_l = P(x_{n+1} = l)$ (i.e., the probability of every instance).

For matrix $\mathbf{Q}_{n+1}^{(t)}$ we have the following update rules:

$$\begin{aligned}\tilde{\mathbf{Q}}_{n+1}^{(t)} &= \mathbf{Q}_{n+1}{}^{(t-1)} \bullet \frac{\mathbf{P}_{(n+1)}}{\mathbf{R}_{(n+1)}^{(t-1)}}\left[\mathbf{Z}_{(n+1)}^{(t-1)}\right]^T \\ \mathbf{Q}_{n+1}^{(t)} &= \frac{\tilde{\mathbf{Q}}_{n+1}^{(t)}}{\tilde{\mathbf{Q}}_{n+1}^{(t)}\mathbf{E}}.\end{aligned} \tag{110}$$

7 Experimental results

We demonstrate the usefulness of the proposed approach using both simulated and real life data.

7.1 Experiments using simulated data

We demonstrate the power of PLTVA framework providing an empirical verification of the fact that the algorithm given by update rules (97) and (98) can produce better bases in terms of interpretability and sparseness than PLSA and its effect on the success of recreating the underlying model. One dataset that has been used to this end is the Swimmer database. The Swimmer database has been used in Donoho and Stodden (2004) for demonstrating a case when the PLTVA can provide a unique decomposition into parts. Some of the images of the Swimmer dataset can be seen in Fig. 2a. The Swimmer image set is comprised of 256 images of dimensions 32×32 forming a tensor $\mathcal{X} \in \Re_{+}^{32\times 32\times 256}$. Each image contains a "torso" (the invariant part) of 12 pixels in the center and four "limbs" of 6 pixels that can be in one of 4 positions. We applied the NMF algorithm given by update rules (13), (14) and (15) (which is equivalent to PLSA) to matrix $\mathbf{X}_{(3)} \in \Re_{+}^{1024\times 256}$ (matricization of tensor $\mathcal{X}$ in terms of the number of images). Matrix $\mathbf{X}_{(3)}$ is first normalized so as is sums up to one. The resulted bases given by the columns of matrix $\mathbf{Q}_{-}$ are shown in Fig. 2b. As can be seen the NMF scheme (Lee and Seung 2000) correctly resolves the local parts but fails on the torso, which is shown as a "ghost" part in all images [this has been also demonstrated in Donoho and Stodden (2004)].

Next we applied the NTF given by (97) and (98) which is equivalent to PLTVA. The proposed NTF algorithm computed bases (which are given by the outer product $\mathbf{q}_1^l \circ \mathbf{q}_2^l$ and $l = 1, \ldots, K$) that contain a unique factorization and correctly resolve the parts (Fig. 2b). Moreover, we performed eigenanalysis to the bases of NMF (PLSA) and NTF(PLTVA) and the mean number of non-zero eigenvalues for NMF was 31.92 and for NTF was 1. This shows that the proposed NTF correctly decomposed the data tensors to rank-one basic tensors (or independent factors given the latent variable).

7.2 Face verification experiments

The experiments conducted with the XM2VTS database used the protocol described in Messer et al. (1999). The images were aligned semi-automatically according to the eyes' position of each facial image using the eye coordinates. The facial images were down-scaled to a resolution of 64×64 pixels. Histogram equalization was used for

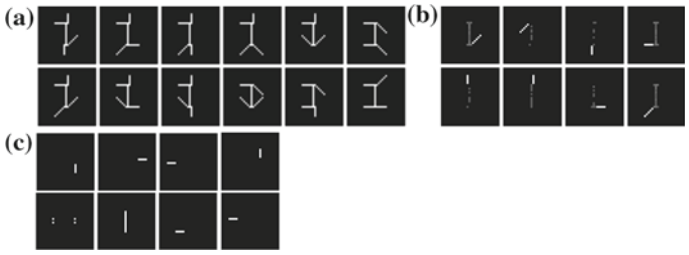

Fig. 2 Some images of the Swimmer database and the corresponding bases for PLSA and PLTVA, **a** Swimmer images; **b** PLSA bases images; **c** PLTVA bases images

the normalization of the facial image luminance and every image was normalized so that is sums up to one.

The XM2VTS database contains 295 subjects, four recording sessions and two shots (repetitions) per recording session. It provides two strict experimental setups, namely, Configuration I and Configuration II (Messer et al. 1999). Each configuration is divided into three different sets: the training set, the validation (in Messer et al. (1999) the set was named evaluation set) and the test set. The training set is used to create client and impostor models for each person. The evaluation set is used to learn the verification decision thresholds. In both the original and the revised manuscript the evaluation set was used in order to learn the parameters of the various methods. That is, the feature dimensionality, kernel parameter and the thresholds were learned using the evaluation set and then the ones which lead to the best EER for every tested method were applied afterwards to the test set. For both configurations, the training set had 200 clients, 25 evaluation impostors and 70 test impostors. The two configurations differed in the distribution of client training and client evaluation data. For additional details concerning the XM2VTS database an interested reader is referred to Messer et al. (1999).

The procedure followed in the experiments was the one also used in Zafeiriou et al. (2006). For comparison reasons the same methodology, using Configuration I of the XM2VTS database, was used. The performance of the algorithms is quoted for the Equal Error Rate (EER) which is the scalar figure of merit that is often used to judge the performance of a verification algorithm. An interested reader is referred to Zafeiriou et al. (2006) and Messer et al. (1999) for more details concerning the XM2VTS protocol and the experimental procedure followed. We have applied both the NMF algorithm which is equivalent to asymmetric PLSA and the proposed NTF algorithm given by (109) and (110) which is the tensor extension of asymmetric PLSA model. In case of NMF we computed the matrix that corresponds to $\mathbb{P}(x_1|z)$ and in case of NTF we computed two matrices on that corresponds to $\mathbb{P}(x_1|z)$ and the other one to $\mathbb{P}(x_2|z)$.

In Zafeiriou (2009a) for extracting features the author projected the images using the bases $\mathbf{q}_1^l \circ \mathbf{q}_2^l$. In this paper we extract features directly from PLSA or PLTVA. That is, we extract features $\mathbb{P}(z|l)$ from a novel image l by applying update rules (110) using as $\mathbb{P}(x_1|z)$ and $\mathbb{P}(x_2|z)$ the matrices that have been computed from the training set (and similar for NMF). Then, the features $P(z|l)$ are used for accepting or rejecting a claim. The best EER using the NMF algorithm that is equivalent to asymmetric PLSA was measured about 2.9% while the best EER for the corresponding NTF algorithm was measured about 1.6% (the best EER for the tested methods is summarized in Table 1). That is, the proposed algorithm resulted to better verification performance. Figure 3 plots the EER as a function of feature dimensionality for the projections based NTF proposed (Zafeiriou 2009a) (abbreviated as NTF–KL–proj), the KL–NTF using as features $P(z|l)$ and the proposed KL–NMF using as features $P(z|l)$, as well.

Table 1 EER for the various tested methods

METHOD	KL–NMF	KL–NTF–Proj.	Proposed KL–NTF
EER	2.9%	3.8%	1.6%

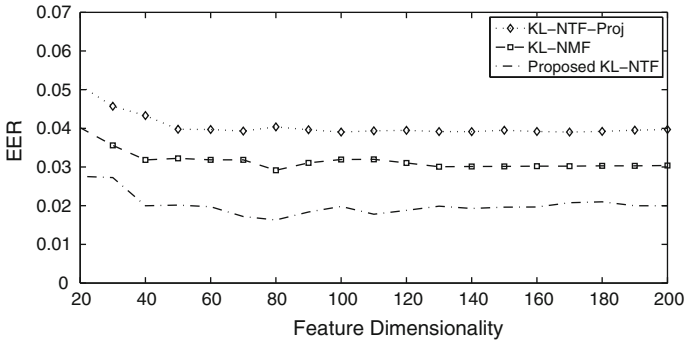

Fig. 3 EER versus feature dimensionality

Table 2 Mean Recognition Error and Variance

METHOD	ORL	FERET
KL–NMF	85.1 (±4.2)	72.6 (±4.7)
KL–NTF–Proj. (Zafeiriou 2009a)	87.85 (±2.7)	84.52 (±3.5)
Proposed KL–NTF	90.2 (±2.9)	86.2 (±2.7)

7.3 Face recognition experiments

Apart of face verification experiments in XM2VTS database we run experiments on ORL and FERET (Phillips et al. 2009, 2000) databases. ORL database consists 40 persons with 10 images each. We used 70 people from the FERET database with 6 images per person. For both databases, half of the images for each person were randomly selected as training set and the remaining as test set. In Table 2 we report the average recognition rate over five random splits of the data and the standard deviation. The test methods were the same applied on XM2VTS for face verification (i.e., NMF, KL–NTF method proposed in Zafeiriou (2009a) and the proposed KL–NTF methods). As can be seen the proposed method achieved the best recognition rate in both databases.

8 Conclusions

In this paper we presented new algorithms for NTF based on Kullback–Leibler divergence. We explored the properties of the optimization problems, investigated the convergence properties and calculated generalized pythagorean rules for KL divergence by formulating the problem as a Csiszar–Tusnady procedure. We explored the relation between the proposed algorithms and probabilistic tensor latent component analysis and generalized over (Finesso and Spreij 2006; Gaussier and Goutte 2005; Ding et al. 2008) and (Shashanka et al. 2008). We proposed decompositions for both symmetric and asymmetric probabilistic latent tensor analysis models. We demonstrate the usefulness of the proposed procedure using both simulated and real life data. We believe

that the proposed algorithms can be used in a variety of applications that span several disciplines such as feature extraction, clustering and classification.

Acknowledgment This work has been supported by the EPSRC project EP/E028659/1 Face Recognition using Photometric Stereo.

9 Appendix

10 Proof of Proposition 2.1

As in Finesso and Spreij (2006), we first prove that there exists a sequence of matrices $(\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_n)$ with $[\mathbf{U}_1\mathbf{e}]_{i_1} = \sum_{i_2=1,\ldots,i_n=1}^{I_2,\ldots,I_n} x_{i_1 i_2 \ldots i_n}$ and $\mathbf{U}_j\mathbf{e} = \mathbf{e}\ \forall\ j \in \{2, \ldots, n\}$ for which $D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right)$ is finite. Now, if we choose $[\mathbf{U}_1]_{i_1 l} = \frac{1}{K} \sum_{i_2=1,\ldots,i_n=1}^{I_2,\ldots,I_n} x_{i_1 i_2 \ldots i_n}$ and for $j \in \{2, \ldots, n\}$ we choose $[\mathbf{U}_j]_{i_j l} = \frac{1}{\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} x_{i_1 \ldots i_n}} \overline{\sum_{i_j}} x_{i_1 \ldots i_{j-1} i_j i_{j+1} \ldots i_n}$. Note that under these conditions $\mathbf{U}_j\mathbf{e} = \mathbf{e}$ and $[\mathbf{U}_1\mathbf{e}]_{i_1} = \overline{\sum_{i_1}} x_{i_1 i_2 \ldots i_n}$ and all the elements $\mathbf{u}_j^l\ \forall\ j = \{1, \ldots, n\}$ and $\bigcirc_{j=1}^{n} \mathbf{u}_j^l$ are nonnegative and finally the cost $D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right)$ is finite.

Next, we shall show that we can restrict ourselves to minimization over a compact set $\mathcal{L}$ of matrices. It will be shown that for positive matrices $\mathbf{U}_1, \ldots, \mathbf{U}_n$, there exist positive matrices $\dot{\mathbf{U}}_1, \ldots, \dot{\mathbf{U}}_n$ with $\left(\dot{\mathbf{U}}_1, \ldots, \dot{\mathbf{U}}_n\right) \in \mathcal{L}$ such that $D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \dot{\mathbf{u}}_j^l\right) \leq D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right)$. We choose arbitrary $\mathbf{U}_1^0, \ldots, \mathbf{U}_n^0$ and we calculate $\mathbf{U}_1^1, \ldots, \mathbf{U}_n^1$ according to the update rules in (36). Hence, $\left[\mathbf{U}_1^1\mathbf{e}\right]_{i_1} = \sum_{i_2=1,\ldots,i_n=1}^{I_2,\ldots,I_n} x_{i_1 i_2 \ldots i_n}$ and $\mathbf{U}_j^1\mathbf{e} = \mathbf{e}\ \forall\ j \in \{2, \ldots, n\}$. Hence, it is sufficient to confine search to the compact set $\mathcal{L}$ where the above constraints are valid.

Fix a set of indices $i_1, \ldots, i_n$. Since we can compute the divergence elementwise, we have the trivial estimate:

$$D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_j^l\right) \geq x_{i_1 \ldots i_n} \log \frac{x_{i_1 \ldots i_n}}{\sum_{l=1}^{K} \prod_{j=1}^{n} u_{i_j j}^l} - x_{i_1 \ldots i_n} + \sum_{l=1}^{K} \prod_{j=1}^{n} u_{i_j j}^l \tag{111}$$

Since for $x_{i_1 \ldots i_n}$ function $d_{i_1 \ldots i_n}(x) : x \to \log \frac{x_{i_1 \ldots i_n}}{x} - x_{i_1 \ldots i_n} + x$ is decreasing on $(0, x_{i_1 \ldots i_n})$, we have for any sufficiently small $x_{i_1 \ldots i_n} < \epsilon < 0$ that $d_{i_1, \ldots, i_n}(x) > d_{i_1 \ldots i_n}(\epsilon)$ for $x \leq \epsilon$ and of course $\lim_{\epsilon \to 0} d_{i_1 \ldots i_n}(\epsilon) = \infty$. Hence, in order to find the minimum of $d_{i_1 \ldots i_n}$, it is sufficient to look at $x \geq \epsilon$. Let $\epsilon_0 > 0$ and such that $\epsilon_0 < \{\min\{x_{i_1 \ldots i_n}\} : x_{i_1 \ldots i_n} > 0\}$. Let $\mathcal{X}$ bet the set of $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_n$ such that $\sum_{l=1}^{K} \prod_{r=1}^{n} u_{i_r, r}^l \geq 0$ for all $i_1, \ldots, i_n$ with $x_{i_1 \ldots i_n} > 0$. Then, the set $\mathcal{X}$ is closed. Take now $\mathcal{K} = \mathcal{L} \cap \mathcal{X}$, then $\mathcal{K}$ is the desired compact set. Let us observe that $\mathcal{K}$ is non-empty for sufficiently small ϵ_0. Clearly, the map $\mathbf{U}_1, \mathbf{U}_2, \ldots, \mathbf{U}_n \to D_{\mathrm{KL}}\left(\mathcal{X} || \sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{u}_{i_j}^l\right)$ is continuous on $\mathcal{K}$ and thus attains its minimum.

11 Proof of

$$D_{\mathrm{KL}}\left(\mathcal{X}||\sum_{l=1}^{K}\prod_{j=1}^{n}\mathbf{u}_j^l\right)=pD_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l\right)+D_{\mathrm{KL}}(p||w_1)$$

Since $\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} p_{i_1\ldots i_n}=1$ and $\sum_{i_1=1,\ldots,i_n=1,l=1}^{I_1,\ldots,I_n,K}\prod_{r=1}^{n}q_{i_r r}^l=1$ we have:

$$\begin{aligned}
&D_{\mathrm{KL}}\left(\mathcal{X}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{u}_j^l\right)\\
&=\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}\left(x_{i_1\ldots i_n}\log\frac{x_{i_1\ldots i_n}}{\sum_{l=1}^{K}\prod_{j=1}^{n}u_{i_j,j}^l}-x_{i_1\ldots i_n}+\sum_{l=1}^{K}\prod_{j=1}^{n}u_{i_j j}^l\right)\\
&=\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}pp_{i_1\ldots i_n}\log\frac{px_{i_1\ldots i_n}}{w_1\sum_{l=1}^{K}\prod_j q_{i_j j}^l}-px_{i_1\ldots i_n}+w_1\sum_{l=1}^{K}\prod_{j=1}^{n}q_{i_j j}^l\\
&=p\left(\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n}p_{i_1\ldots i_n}\log\frac{x_{i_1\ldots i_n}}{\sum_{l=1}^{K}\prod_{j=1}^{n}q_{i_j j}^l}\right)+\left(p\log\frac{p}{w_1}-p+w_1\right)\\
&=pD_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l\right)+D_{\mathrm{KL}}(p||w_1)
\end{aligned}\tag{112}$$

12 Proof of Theorem 2.1

Before we proceed to the proof, one should refer to the proof of Lemma 2.2 for some notation definition that should be used. We first show that $\mathbf{Q}_1^{(t)},\ldots,\mathbf{Q}_n^{(t)}$ form convergent sequences. We start with Eq. 76. By summing over t, we obtain:

$$\begin{aligned}
&D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{Q}^{(0)}\right)-D_{\mathrm{KL}}\left(\mathcal{P}||\mathcal{Q}^{(t)}\right)\\
&=\sum_{k=1}^{(t-1)}\left(D_{\mathrm{KL}}\left(\mathcal{V}^k||\mathcal{V}^{k+1}\right)+D_{\mathrm{KL}}\left(\mathcal{Q}^{(k+1)}||\mathcal{Q}^{(k)}\right)\right)
\end{aligned}\tag{113}$$

It follows that $\sum_{k=1}^{\infty}D_{\mathrm{KL}}\left(\mathcal{V}^{(k)}||\mathcal{V}^{(k+1)}\right)$ and $\sum_{k=1}^{\infty}D_{\mathrm{KL}}\left(\mathcal{Q}^{(t+1)}||\mathcal{Q}^{(t)}\right)$ are finite. Now by using the Hellinger distance, $H\left(\mathbb{P},\mathbb{Q}\right)\leq D_{\mathrm{KL}}\left(\mathbb{P}||\mathbb{Q}\right)$ (Shiryaev 1996). In our case we have $H\left(\mathbb{Q}^{(k)},\mathbb{Q}^{(k+1)}\right)=\sum_{i_1=1,l=1,i_2=1,\ldots,i_n=1}^{I_1,K,I_2,\ldots,I_n}(\sqrt{\mathcal{Q}^{(k)}(i_1,l,i_2,\ldots,i_n)}-\sqrt{\mathcal{Q}^{(k+1)}(i_1,l,i_2,\ldots,i_n)})^2$. So we obtain that:

$$\sum_{k=1}^{\infty}H\left(\mathbf{Q}^{(k)},\mathbf{Q}^{(k+1)}\right)<\infty\tag{114}$$

We therefore have that, pointwise, tensor $\mathcal{Q}^{(t)}$ forms a Caushy sequence and hence it has a limit $\mathcal{Q}^{\infty}$. We shall show that $\mathcal{Q}^{\infty}$ belongs $\mathcal{Q}$. Before we proceed, let us define the following notation. $\mathcal{Q}(i_j,l)=\sum_{i_1=1,\ldots,i_{j-1}=1,i_{j+1}=1,\ldots,i_n=1}^{I_1,\ldots,I_{j-1},I_{j+1},\ldots,I_n}\mathcal{Q}(i_1,l,i_2,\ldots,i_n)$, $\mathcal{Q}(l)$

$= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathcal{Q}(i_1, l, i_2, \ldots, i_n)$ and $\mathcal{Q}\left(i_1, l, i_2, \ldots, i_{j-1}, i_{j+1}, \ldots, i_n\right) = \sum_{i_j=1}^{I_j} \mathcal{Q}\left(i_1, l, i_2, \ldots, i_n\right)$. This notation will be used for other tensors as well.

Since $\mathcal{Q}^{(t)}(i_1, l, i_2, \ldots, i_n)$ converges to limits $\mathcal{O}^{\infty}(i_1, l, i_2, \ldots, i_n)$, by summation, we have that the marginals $\mathbf{Q}_1^{(t)}(i_1, l) = \mathcal{Q}^{(t)}(i_1, l)$ converge to limits $\mathcal{Q}^{\infty}(i_1, l)$ and likewise we have convergence of the marginals $\mathcal{Q}^{(t)}(i_j, l)$ to $\mathcal{Q}^{\infty}(i_j, l)$ and $\mathcal{Q}^{(t)}(l)$ to $\mathcal{Q}^{\infty}(l)$. Hence, if $\mathcal{Q}^{\infty}(l) > 0$, then the $\mathbf{Q}_j^{\infty}(i_j, l)$ converge to $\mathbf{Q}_j^{\infty}(i_j, l) = \frac{\mathcal{Q}^{\infty}(i_j, l)}{\mathcal{Q}^{\infty}(l)}$ and we have $\mathcal{Q}^{\infty}(i_1, l, i_2, \ldots, l, i_n) = \mathcal{Q}^{\infty}(i_1, l, i_2, \ldots, i_{j-1}, i_{j+1}, \ldots, l, i_n)$ $\mathbf{Q}_j^{\infty}(i_j, l)$ for $j = 2, \ldots, n$. Now for the case where $\mathcal{Q}^{\infty}(l_0) = 0$ for some l_0. In this case both $\mathcal{Q}(i_1, l, i_2, \ldots, i_n)$ and $\mathcal{Q}^{\infty}(i_1, l)$ are zero, but still we may have a factorization $\mathcal{Q}^{\infty}(i_1, l, i_2, \ldots, i_n) = \mathbf{Q}_1^{\infty}(i_1, l_0)\mathbf{Q}_2^{\infty}(i_2, l_0) \ldots \mathbf{Q}_n^{\infty}(i_n, l_0)$ where we may assign arbitrary values to $\mathbf{Q}_j^{\infty}(i_j, l_0)$ for $j = \{2, \ldots, n\}$. Let $\mathcal{L}$ be the set of l for which $\sum_{i_1=1}^{I_1} \mathbf{Q}_1^{\infty}(i_1, l) > 0$. Then, $\mathcal{O}^{\infty}(i_1, \ldots, i_n) = \sum_{l \in \mathcal{L}} \mathcal{O}^{\infty}(i_1, l) \ldots \mathcal{O}^{\infty}(i_n, l)$ and the $\mathcal{O}^{(t)}$ converge to $\mathcal{O}^{\infty}$ and the theorem is proven.

13 Proof of Theorem 2.2

By computing the first order partial derivatives of the objective function, using the middle term of Eq. 45, we can rewrite the update Eqs. 48 and 49 as:

$$\mathbf{Q}_1^{(t+1)}(i_1, l) = \mathbf{Q}_1^{(t)}(i_1, l)\left(-\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial \mathbf{Q}_1(i_1, l)} + 1\right) \tag{115}$$

and for $j = 2, \ldots, n$

$$\begin{aligned} &\mathbf{Q}_j^{(t+1)}(i_j, l)\left(\sum_{i_1=1}^{I_1} \mathbf{Q}_1^{(t+1)}(i_1, l)\right) \\ &\quad = \mathbf{Q}_j(i_j, l)\left(-\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial \mathbf{Q}_j(i_j, l)} + \sum_{i_1=1}^{I_1} \mathbf{Q}_1^{(t+1)}(i_1, l)\right) \end{aligned} \tag{116}$$

where $\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial \mathbf{Q}_1(i_1, l)}$ stands for the partial derivative evaluated at $(\mathbf{Q}_1^{(t)}, \ldots, \mathbf{Q}_n^{(t)})$ and likewise $\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial \mathbf{Q}_j(i_j, l)}$.

Let $(\mathbf{Q}_1, \ldots, \mathbf{Q}_n)$ be a limit point of the algorithm (i.e., $\mathbf{Q}_j^{(t+1)} = \mathbf{Q}_j^{(t)}$ for all j). Equations 115 and 116 become:

$$\mathbf{Q}_1(i_1, l) = \mathbf{Q}_1(i_1, l)\left(-\frac{\partial D_{\mathrm{KL}}}{\partial \mathbf{Q}_1(i_1, l)} + 1\right) \tag{117}$$

and

$$\mathbf{Q}_j(i_j,l)\left(\sum_{i_1=1}^{I_1}\mathbf{Q}_1(i_1,l)\right)=\mathbf{Q}_j(i_j,l)\left(-\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial\mathbf{Q}_j(i_j,l)}+\sum_{i_1=1}^{I_1}\mathbf{Q}_1(i_1,l)\right) \quad (118)$$

which implies:

$$\mathbf{Q}_j(i_j,l)\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial\mathbf{Q}_j(i_j,l)}=0 \quad (119)$$

for all $j=1,\ldots,n$.

First, we start with $\mathbf{Q}_1$. Suppose that for some i_1 and l we have $\mathbf{Q}_1(i_1,l)>0$, then necessarily $\frac{\partial D_{\mathrm{KL}}}{\partial\mathbf{Q}_1(i_1,l)}=0$. Now, suppose for a pair i_1 and l we have $\mathbf{Q}_1(i_1,l)=0$ and $\frac{\partial D_{\mathrm{KL}}}{\partial\mathbf{Q}_1(i_1,l)}<0$. Of course, by continuity, this partial derivative will be negative in a sufficiently small neighborhood of this limit point. Since, we deal with a limit point of the algorithm, we must have infinitely often $\mathbf{Q}_1^{(t+1)}(i_1,l)<\mathbf{Q}_1^{(t)}(i_1,l)$. From (117), we then conclude that in these points we do have $\frac{\partial D_{\mathrm{KL}}}{\partial\mathbf{Q}_1(i_1,l)}>0$. Clearly, this contradicts our assumption of a negative partial derivative. Hence, we conclude that $\frac{\partial D_{\mathrm{KL}}}{\partial\mathbf{Q}_1(i_1,l)}\geq 0$ when $\mathbf{Q}_1(i_1,l)=0$.

For $j=2,\ldots,n$, let $\mathbf{Q}_j(i_j,l)>0$. Then the corresponding partial derivative is zero. Let l be such that $\mathbf{Q}_j(i_j,l)=0$ and suppose that we have $\frac{\partial D_{\mathrm{KL}}}{\partial\mathbf{Q}_j(i_j,l)}<0$. If we run the algorithm, then $\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial\mathbf{Q}_j(i_j,l)}/\sum_{i_1=1}^{I_1}\mathbf{Q}_1^{(t+1)}(i_1,l)$ converges to a negative limit, whereas $\sum_{i_1=1}^{I_1}\mathbf{Q}_1(i_1,l)/\sum_{i_1=1}^{I_1}\mathbf{Q}_1^{(t+1)}(i_1,l)$ converges to one. Hence, there is $\eta>0$ such that $\frac{\partial D_{\mathrm{KL}}}{\partial\mathbf{Q}_j(i_j,l)}<-2\eta/3$ and $\sum_{i_1=1}^{I_1}\mathbf{Q}_1(i_1,l)/\sum_{i_1=1}^{I_1}\mathbf{Q}_1^{(t+1)}(i_1,l)>1-\eta/3$. Hence, eventually we shall have from (118):

$$\mathbf{Q}_j^{(t+1)}(i_j,l)-\mathbf{Q}_j^{(t)}(i_j,l)=\mathbf{Q}_j^{(t+1)}\left(-\frac{\frac{\partial D_{\mathrm{KL}}^{(t)}}{\partial\mathbf{Q}_j(i_j,l)}}{\sum_{i_1=1}^{I_1}\mathbf{Q}_1^{(t+1)}(i_1,l)}+\frac{\sum_{i_1=1}^{I_1}\mathbf{Q}_1(i_1,l)}{\sum_{i_1=1}^{I_1}\mathbf{Q}_1^{(t+1)}(i_1,l)}\right)>\eta/3 \quad (120)$$

which contradicts with the convergence of $\mathbf{Q}_j^{(t)}(i_j,l)$ to zero.

This theorem says nothing about the convergence of $\mathbf{Q}_j(i_j,l)$ for those l such that $\sum_{i_1=1}^{I_1}\mathbf{Q}_1^{\infty}(i_1,l)=0$. But this behavior is uninteresting from a factorization point of view. If the lth column of $\mathbf{Q}_1^{\infty}$ is zero, the values of the lth column of $\mathbf{Q}_j^{\infty}$ are not relevant, since they don't appear in the product $\sum_{m=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l$. This case corresponds to a factorization with less than K rank-one tensors.

14 Proof of Corollary 2.1

Consider the Lagrangian function L defined by

$$L(\mathbf{Q}_1, \ldots, \mathbf{Q}_n) = D_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l\right) - \sum_{i_1=1,l=1}^{I_1,K}\boldsymbol{\Lambda}_1(i_1,l)\mathbf{Q}_1(i_1,l) - \cdots - \sum_{i_n=1,l=1}^{I_n,K}\boldsymbol{\Lambda}_n(i_n,l)\mathbf{Q}_n(i_n,l) \tag{121}$$

where $\boldsymbol{\Lambda}_j$ is a sequence of n matrices of Lagrangian multipliers. Let us concentrate on partial derivative $\frac{\partial L}{\partial \mathbf{Q}_1(i_1,l)}$ in a fixed point of the algorithm. From Theorem 2.2, it is known that $\mathbf{Q}_1(i_1,l)\frac{\partial D_{\mathrm{KL}}}{\partial \mathbf{Q}_1} = 0$. Suppose that $\mathbf{Q}_1(i_1,l) > 0$, then $\frac{\partial D_{\mathrm{KL}}}{\mathbf{Q}_1(i_1,l)} = 0$ and the KKT conditions on for this variable are satisfied with $\boldsymbol{\Lambda}(i_1,l) = 0$. In case $\mathbf{Q}_1(i_1,l) = 0$, we know from Theorem 2.2 that $\frac{\partial D_{\mathrm{KL}}}{\partial \mathbf{Q}_1(i_1,l)} \geq 0$. By choosing $\boldsymbol{\Lambda}(i_1,l) = \frac{\partial D_{\mathrm{KL}}}{\partial \mathbf{Q}_1(i_1,l)} \geq 0$, we see that the Karush–Kuhn-Tucker conditions are satisfied.

15 Proof of Lemma 2.1

If $\wp \cap \Im \neq \varnothing$ then there exist a tensor $\mathcal{O} \in \aleph$ which also belongs to $\wp$, therefore $\mathcal{P} = \sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l$. Conversely, if we have $\mathcal{P} = \sum_{l=1}^{K}\bigcirc_{j=1}^{n}\mathbf{q}_j^l$ with K rank-one-tensors, then tensor $\mathcal{V}(i_1,l,i_2,\ldots,i_n) = \prod_{j=1}^{K}\mathbf{Q}_j(i_j,l)$ clearly belongs to $\wp$. Without loss of generality, we assume that $\mathbf{Q}_j\mathbf{e} = \mathbf{e}$, so that $\mathcal{V}$ belongs to $\aleph$ as well.

16 Proof of Proposition 2.2

With $\mathcal{V}^{(\mathrm{opt})} = \mathcal{V}^{(\mathrm{opt})}(\mathcal{Q})$, being the optimal solution of the partial minimization over $\mathcal{P}$, we have:

$$\begin{aligned} D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q}) &\geq D_{\mathrm{KL}}\left(\mathcal{V}^{(\mathrm{opt})}||\mathcal{Q}\right) \\ &= D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O}) \\ &\geq \min_{\mathcal{O}\in\aleph} D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O}). \end{aligned} \tag{122}$$

It follows that $\inf_{\mathcal{V}\in\wp,\mathcal{Q}\in\Im} D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q}) \geq \min_{\mathcal{O}\in\aleph} D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O})$.

Conversely, let $\mathcal{Q} \in \Im$ be given and let $\mathcal{O}$ be defined as $\mathcal{O}(i_1,\ldots,i_n) = \sum_{l=1}^{K}\mathcal{Q}(i_1,\ldots,l,i_n)$. Now from

$$\begin{aligned} D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O}) &= D_{\mathrm{KL}}\left(\mathcal{V}^{(\mathrm{opt})}(\mathcal{Q})||\mathcal{Q}\right) \\ &\geq \inf_{\mathcal{V}\in\wp,\mathcal{Q}\in\Im} D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q}), \end{aligned} \tag{123}$$

we obtain

$$\min_{\mathcal{O}\in\aleph} D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O}) \geq \inf_{\mathcal{V}\in\wp,\,\mathcal{Q}\in\Im} D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q}) \tag{124}$$

Finally, we show that the infima can be replaced by minima. Let $\mathbf{Q}_1^{(\mathrm{opt})}, \mathbf{Q}_2^{(\mathrm{opt})}, \ldots, \mathbf{Q}_n^{(\mathrm{opt})}$ be such that $(\mathbf{Q}_1, \ldots, \mathbf{Q}_n) \rightarrow D_{\mathrm{KL}}\left(\mathcal{P}||\sum_{l=1}^{K} \bigcirc_{j=1}^{n} \mathbf{q}_j^l\right)$ is minimized (their existence is guaranteed by Proposition 2.1).

17 Proof of Lemma 2.2

To prove Pythagorean rule (63) we start by expanding $D_{\mathrm{KL}}\left(\mathcal{V}||\mathcal{V}^{(\mathrm{opt})}\right)$ and $D_{\mathrm{KL}}\left(\mathcal{V}^{(\mathrm{opt})}||\mathcal{Q}\right)$ using the $\mathcal{P}^{(\mathrm{opt})}$ from Eq. 57 and $\sum_{l=1}^{K} \mathcal{V}(i_1, \ldots, i_{n-1}, l, i_n) = \mathcal{P}(i_1, \ldots, i_n)$:

$$\begin{aligned}
&D_{\mathrm{KL}}\left(\mathcal{V}||\mathcal{V}^{(\mathrm{opt})}\right) + D_{\mathrm{KL}}\left(\mathcal{V}^{(\mathrm{opt})}||\mathcal{Q}\right) \\
&= \sum\nolimits_{i_1=1,l=1,i_2=1,\ldots,i_n=1}^{I_1,K,I_2,\ldots,I_n} \mathcal{V}(i_1, l, i_2, \ldots, i_n) \log \frac{\mathcal{V}(i_1, l, i_2, \ldots, i_n)\mathcal{O}(i_1, \ldots, i_n)}{\mathcal{Q}(i_1, \ldots, i_n)\mathcal{P}(i_1, \ldots, i_n)} \\
&\quad + \sum\nolimits_{i_1=1,l=1,i_2=1,\ldots,i_n=1}^{I_1,K,I_2,\ldots,I_n} \mathcal{Q}(i_1, l, i_2, \ldots, i_n) \frac{\mathcal{P}(i_1, \ldots, i_n)}{\mathcal{O}(i_1, \ldots, i_n)} \log \frac{\mathcal{P}(i_1, \ldots, i_n)}{\mathcal{O}(i_1, \ldots, i_n)} \\
&= \sum\nolimits_{i_1=1,l=1,i_2=1,\ldots,i_n=1}^{I_1,K,I_2,\ldots,I_n} \mathcal{V}(i_1, l, i_2, \ldots, i_n) \log \frac{\mathcal{V}(i_1, l, i_2, \ldots, i_n)}{\mathcal{Q}(i_1, l, i_2, \ldots, i_n)} \\
&\quad + \sum\nolimits_{i_1=1,l=1,i_2=1,\ldots,i_n=1}^{I_1,K,I_2,\ldots,I_n} \mathcal{V}(i_1, l, i_2, \ldots, i_n) \log \frac{\mathcal{O}(i_1, \ldots, i_n)}{\mathcal{P}(i_1, \ldots, i_n)} \\
&\quad + \sum\nolimits_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathcal{O}(i_1, \ldots, i_n) \frac{\mathcal{P}(i_1, \ldots, i_n)}{\mathcal{O}(i_1, \ldots, i_n)} \log \frac{\mathcal{P}(i_1, \ldots, i_n)}{\mathcal{O}(i_1, \ldots, i_n)} = D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O})
\end{aligned} \tag{125}$$

To prove the relation (64) insert Eq. 57 into $D_{\mathrm{KL}}(\mathcal{V}^{(\mathrm{opt})}||\mathcal{Q})$ and sum over l to get:

$$\begin{aligned}
&D_{\mathrm{KL}}(\mathcal{V}^{(\mathrm{opt})}||\mathcal{Q}) \\
&= \sum_{i_1=1,l=1,i_2=1,\ldots,i_n=1}^{I_1,K,I_2,\ldots,I_n} \mathcal{P}(i_1 = 1, \ldots, i_n = 1) \frac{\mathcal{Q}(i_1, l, i_2, \ldots, i_n)}{\mathcal{O}(i_1, \ldots, i_n)} \log \frac{\mathcal{P}(i_1, \ldots, i_n)}{\mathcal{O}(i_1, \ldots, i_n)} \\
&= D_{\mathrm{KL}}(\mathcal{V}||\mathcal{Q})
\end{aligned} \tag{126}$$

To prove the Pythagorian rule (66) we should introduce the following notation. Let $\mathbf{V}(i_1, l) = \sum_{i_2=1,\ldots,i_n=1}^{I_2,\ldots,I_n} \mathcal{V}(i_1, l, i_2, \ldots, i_n)$, for $j = \{2, \ldots, n\}$ then $\mathbf{V}(i_j, l) = \sum_{i_1=1,\ldots,i_{j-1},i_{j+1},\ldots,I_n}^{I_1,\ldots,I_{j-1},I_{j+1},\ldots,I_n} \mathbf{V}(i_1, l, i_2, \ldots, i_n)$. Moreover, for $j = \{2, \ldots, n\}$ we introduce $\mathbf{V}(i_j|l) = \frac{\mathbf{V}(i_j,l)}{\sum_{i_j=1}^{I_j} \mathcal{V}(i_j,l)}$. For $\mathcal{Q}$ we introduce the similar notation and observe

that $\mathbf{Q}(i_1,l)=\mathbf{Q}_1(i_1,l)$, $\mathbf{Q}_j(i_j|l)=\frac{\mathbf{Q}_j(i_j,l)}{\sum_{i_j=1}^{I_j}\mathbf{Q}_j(i_j,l)}$, $\mathbf{Q}_1^{(\text{opt})}(i_1,l)=\mathbf{V}(i_1,l)$ with $j=\{2,\ldots,n\}$ and $\mathbf{Q}_j^{(\text{opt})}(i_j,l)=\mathbf{V}(i_j|l)$.

Now we compute:

$$\begin{aligned}
&D_{\text{KL}}(\mathcal{V}||\mathcal{Q})-D_{\text{KL}}\left(\mathcal{V}||\mathcal{Q}^{(\text{opt})}\right)\\
&=\sum\nolimits_{i_1=1,l=1,i_2=1,\ldots,i_n=1}^{I_1,K,I_2,\ldots,I_n}\mathcal{V}(i_1,l,i_2,\ldots,i_n)\left(\log\frac{\mathbf{V}(i_1,l)}{\mathbf{Q}_1(i_1,l)}+\log\frac{\mathbf{V}(i_2|l)}{\mathbf{Q}_1(i_1,l)}\right.\\
&\quad\left.+\log\frac{\mathbf{V}(i_2|l)}{\mathbf{Q}_2(i_2,l)}+\cdots+\log\frac{\mathbf{V}(i_n|l)}{\mathbf{Q}_n(i_n,l)}\right)\\
&=\sum\nolimits_{i_1=1,l=1}^{K}\mathbf{V}(i_1,l)\log\frac{\mathbf{V}(i_1,l)}{\mathbf{Q}_1(i_1,l)}+\sum\nolimits_{i_2=1,l=1}^{I_2,K}\mathbf{V}(i_2,l)\log\frac{\mathbf{V}(i_2|l)}{\mathbf{Q}_2(i_2,l)}+\cdots\\
&\quad+\sum\nolimits_{i_n=1,l=1}^{I_n,K}\mathbf{V}(i_n,l)\log\frac{\mathbf{V}(i_n|l)}{\mathbf{Q}_n(i_n,l)}\\
&=\sum\nolimits_{i_1=1,l=1}^{I_1,K}\mathbf{V}(i_1,l)\log\frac{\mathbf{V}(i_1,\ldots,l)}{\mathbf{Q}_1(i_1,l)}+\sum\nolimits_{i_2=1,l=1}^{I_2,K}\mathbf{V}(i_2,l)\log\frac{\mathbf{V}(i_2|l)}{\mathbf{Q}_2(i_2,l)}+\cdots\\
&\quad+\sum\nolimits_{i_n=1,l=1}^{I_2,K}\mathbf{V}(i_n,l)\log\frac{\mathbf{V}(i_n|l)}{\mathbf{Q}_n(i_n,l)}\\
&=D_{\text{KL}}\left(\mathcal{Q}^{(\text{opt})}||\mathcal{Q}\right)
\end{aligned}\tag{127}$$

18 Generalized Pythagorean rule in (72)

We start from Eq. 70 Lemma 2.3

$$\begin{aligned}
\mathbb{E}_{\mathbb{P}}D_{\text{KL}}\left(\mathbb{P}^{U|V}||\mathbb{Q}^{U|V}\right)&=\mathbb{E}_{\mathbb{P}}D_{\text{KL}}\left(\mathbb{P}^{U|V}||\mathbb{P}^{U|V_1}\right)\\
&\quad+\mathbb{E}_{\mathbb{P}}D_{\text{KL}}\left(\mathbb{P}^{U|V_1}||\mathbb{Q}^{U|V_1}\right)
\end{aligned}\tag{128}$$

by letting $V_2=Y_1$, $U=Y_2,\ldots,Y_n$ and $V_1=X$ and replacing in Eq. 71 then

$$\begin{aligned}
D_{\text{KL}}\left(\mathcal{V}||\mathcal{Q}\right)&=\mathbb{E}_{\mathbb{P}}D_{\text{KL}}\left(\mathbb{P}^{Y_2,\ldots,Y_n|X,Y_1}||P^{Y_2,\ldots,Y_n|X}\right)\\
&\quad+\mathbb{E}_{\mathbb{P}}D_{\text{KL}}\left(P^{Y_2,\ldots,Y_n|X}||\mathbb{Q}^{Y_2,\ldots,Y_n|X}\right)\\
&\quad+D_{\text{KL}}\left(\mathbb{P}^{Y_1,X}||\mathbb{Q}^{Y_1,X}\right)
\end{aligned}\tag{129}$$

now for

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}}D_{\text{KL}}\left(\mathbb{P}^{Y_2,\ldots,Y_n|X}||\mathbb{Q}^{Y_2,\ldots,Y_n}\right)\\
&=\mathbb{E}_{\mathbb{P}}\sum\mathbb{P}^{Y_2,\ldots,Y_n|X}\left(Y_2=i_2,\ldots,Y_n=i_n|X=l\right)
\end{aligned}$$

$$
\begin{aligned}
&\times \log \frac{\mathbb{P}^{Y_2,\ldots,Y_n}(Y_2 = i_2, \ldots, Y_n = i_n | X = l)}{\mathbb{Q}^{Y_2,\ldots,Y_n|X}(Y_2 = i_2, \ldots, Y_n = i_n | X = l)} \\
&= \mathbb{E}_{\mathbb{P}} \sum \mathbb{P}(Y_2 = i_2 | X = l) \ldots \mathbb{P}(i_n = i_n | X = l) \\
&\quad \times \log \frac{\mathbb{P}(Y_2 = i_2 | X = l) \ldots \mathbb{P}(Y_n = i_n | X = l)}{\mathbb{Q}(Y_2 = i_2 | X = l) \ldots \mathbb{Q}(Y_n = i_n | X = l)} \\
&= \mathbb{E}_{\mathbb{P}} \sum_{i_2=1,\ldots,i_n=1}^{I_2,\ldots,I_n} \mathbb{P}(Y_2 = i_2 | X = l) \ldots \mathbb{P}(i_n = i_n | X) \\
&\quad \times \left(\log \frac{\mathbb{P}(Y_2 = i_2 | X = l)}{\mathbb{Q}(Y_2 = i_2 | X = l)} + \cdots + \log \frac{\mathbb{P}(Y_n = i_n | X = l)}{\mathbb{Q}(Y_n = i_n | X = l)} \right) \\
&= \mathbb{E}_{\mathbb{P}} \left(D_{\mathrm{KL}}(\mathbb{P}^{Y_2|X} || \mathbb{Q}^{Y_2|X}) + \cdots + D_{\mathrm{KL}}(\mathbb{P}^{Y_n|X} || \mathbb{Q}^{Y_n|X}) \right) \\
&= \sum_{i=2,\ldots,n} \mathbb{E}_{\mathbb{P}} D_{\mathrm{KL}} \left(\mathbb{P}^{Y_i|X} || \mathbb{Q}^{Y_i|X} \right)
\end{aligned} \tag{130}
$$

19 Proof of Proposition 2.3

The two Pythagorean rules from Lemma 2.2 now take the forms

$$
\begin{aligned}
D_{\mathrm{KL}} \left(\mathcal{V}^{(t)} || \mathcal{Q}^{(t)} \right) &= D_{\mathrm{KL}} \left(\mathcal{V}^{(t)} || \mathcal{Q}^{(t+1)} \right) + D_{\mathrm{KL}} \left(\mathcal{Q}^{(t+1)} || \mathcal{Q}^{(t)} \right) \\
D_{\mathrm{KL}} \left(\mathcal{V}^{(t)} || \mathcal{Q}^{(t+1)} \right) &= D_{\mathrm{KL}} \left(\mathcal{V}^{(t)} || \mathcal{V}^{(t+1)} \right) + D_{\mathrm{KL}} \left(\mathcal{V}^{(t+1)} || \mathcal{V}^{(t)} \right)
\end{aligned} \tag{131}
$$

By adding these two equations results in

$$
\begin{aligned}
D_{\mathrm{KL}}(\mathcal{V}^{(t)} || \mathcal{Q}^{(t)}) &= D_{\mathrm{KL}}(\mathcal{V}^{(t)} || \mathcal{V}^{(t+1)}) + D_{\mathrm{KL}}(\mathcal{V}^{(t+1)} || \mathcal{Q}^{(t+1)}) \\
&\quad + D_{\mathrm{KL}}(\mathcal{Q}^{(t+1)} || \mathcal{Q}^{(t)})
\end{aligned} \tag{132}
$$

and since $D_{\mathrm{KL}}(\mathcal{V}^{(t)} || \mathcal{Q}^{(t)}) = D_{\mathrm{KL}}(\mathcal{P} || \mathcal{O}^{(t)})$ from (64) the result follows.

A practical result of the above is the following. If one starts the algorithms with matrices $(\mathbf{Q}_1^0, \ldots, \mathbf{Q}_n^0)$ in the interior of the domain, the iterations will remain in the interior. Suppose that, at step n, the update gain is zero. Then, we get that $D_{\mathrm{KL}}(\mathcal{Q}^{(t+1)} || \mathcal{Q}^{(t)}) = 0$. Hence the tensors $\mathcal{Q}^{(t+1)}$ and $\mathcal{Q}^{(t)}$ are identical. From this it follows by summation that $\mathbf{Q}_1^{(t+1)} = \mathbf{Q}_1^{(t)}$. But, then we also have the equality for $j \neq 2$

$$
\begin{aligned}
&\mathbf{Q}_1^{(t)}(i_1, l) \ldots \mathbf{Q}_{j-1}^{(t)}(i_{j-1}, l) \mathbf{Q}_j^{(t+1)}(i_j, l) \mathbf{Q}_{j+1}^{(t)}(i_{j+1}, l) \ldots \mathbf{Q}_n^{(t)}(i_n, l) \\
&= \mathbf{Q}_1^{(t)}(i_1, l) \ldots \mathbf{Q}_{j-1}^{(t)}(i_{j-1}, l) \mathbf{Q}_j^{(t+1)}(i_j, l) \mathbf{Q}_{j+1}^{(t)}(i_{j+1}, l) \ldots \mathbf{Q}_n^{(t)}(i_n, l)
\end{aligned} \tag{133}
$$

for all $i_1, \ldots, i_n$ and l. Since $\mathbf{Q}_1^{(t)}(i_1, l)$ are positive and summing for all $r = \{2, \ldots, n\} - \{j\}$ then we have $\mathbf{Q}_j^{(t+1)} = \mathbf{Q}_j^{(t)}$. Hence, the updating formulas strictly decrease the objective function until the algorithm reaches a fixed point.

20 Proof of Lemma 2.4

We can start from Eq. 79

$$W\left(\mathbf{Q}_1, \ldots, \mathbf{Q}_n, \mathbf{Q}'_1, \ldots, \mathbf{Q}'_n\right) = D_{\mathrm{KL}}\left(\mathcal{P}||\mathbb{Q}'^{Y}\right) + \mathbb{E}_{\mathbb{P}} D_{\mathrm{KL}}\left(\mathbb{Q}^{X|Y}||\mathbb{Q}^{X|Y'}\right) \quad (134)$$

since $\mathbb{Q}^{X|Y}\left(Y_1 = i_1, \ldots, Y_n = i_n\right) = \frac{\mathcal{Q}(Y_1=i_1, X=l, Y_2=i_2, \ldots, Y_n=i_n)}{\mathcal{O}(Y_1=i_1, \ldots, Y_n=i_n)}$ we can expand as:

$$\begin{aligned}
&\mathbb{E}_{\mathbb{P}} D_{\mathrm{KL}}\left(\mathbb{Q}^{X|Y}||\mathbb{Q}^{X|Y'}\right) \\
&= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}\left(i_1, \ldots, i_n\right) \sum_{l=1}^{K} \frac{\mathcal{Q}\left(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n\right)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\
&\quad \times \log \frac{\frac{\mathcal{Q}(Y_1=i_1,\ldots,Y_n=i_n,X=l)}{\mathcal{O}(Y_1=i_1,\ldots,Y_n=i_n)}}{\frac{\mathcal{Q}'(Y_1=i_1,\ldots,Y_n=i_n,X=l)}{\mathcal{O}'(Y_1=i_1,\ldots,Y_n=i_n)}} \\
&= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \sum_{l=1}^{K} \frac{\mathcal{Q}(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\
&\quad \times \left(\log \frac{\mathcal{Q}(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n)}{\mathcal{Q}'(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n)}\right. \\
&\quad \left. + \log \frac{\mathcal{O}'(Y_1 = i_1, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)}\right) \\
&= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1, l) \ldots \mathbf{Q}_n(i_n, l)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\
&\quad \times \log \frac{\mathbf{Q}_1(i_1, l) \ldots \mathbf{Q}_n(i_n, l)}{\mathbf{Q}'_1(i_1, l) \ldots \mathbf{Q}'_n(i_n, l)} \\
&\quad + \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \sum_{l=1}^{K} \frac{\mathcal{Q}(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\
&\quad \times \log \frac{\mathcal{O}'(Y_1 = i_1, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)}
\end{aligned} \quad (135)$$

then

$$A_2 = \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \sum_{l=1}^{K} \frac{\mathcal{Q}(Y_1 = i_1, X = l, Y_2 = i_2, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)}$$

$$\times \log \frac{\mathcal{O}'(Y_1 = i_1, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)}$$
$$= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \log \frac{\mathcal{O}'(Y_1 = i_1, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \tag{136}$$

and

$$\begin{aligned} D_{\mathrm{KL}}\left(\mathcal{P}||\mathbb{Q}'^{Y}\right) + A_2 &= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \log \frac{\mathbb{P}(i_1, \ldots, i_n)}{\mathcal{O}'(Y_1 = i_1, \ldots, Y_n = i_n)} \\ &\quad + \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \log \frac{\mathcal{O}'(Y_1 = i_1, \ldots, Y_n = i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\ &= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \log \frac{\mathbb{P}(i_1, \ldots, i_n)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\ &= D_{\mathrm{KL}}(\mathcal{P}||\mathcal{O}) \end{aligned} \tag{137}$$

For the other term

$$\begin{aligned} -A_1 &= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1, l) \ldots \mathbf{Q}_n(i_n, l)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\ &\quad \times \log \frac{\mathbf{Q}'_1(i_1, l) \ldots \mathbf{Q}'_n(i_n, l)}{\mathbf{Q}_1(i_1, l) \ldots \mathbf{Q}_n(i_n, l)} \\ &= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1, l) \ldots \mathbf{Q}_n(i_n, l)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \\ &\quad \times \left(\log \frac{\mathbf{Q}'_1(i_1, l)}{\mathbf{Q}_1(i_1, l)} + \ldots + \log \frac{\mathbf{Q}'_n(i_n, l)}{\mathbf{Q}_n(i_n, l)} \right) \\ &= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \\ &\quad \times \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1, l) \ldots \mathbf{Q}_n(i_n, l)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \log \frac{\mathbf{Q}'_1(i_1, l)}{\mathbf{Q}_1(i_1, l)} + \ldots \\ &\quad + \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1, \ldots, i_n) \\ &\quad \times \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1, l) \ldots \mathbf{Q}_n(i_n, l)}{\mathcal{O}(Y_1 = i_1, \ldots, Y_n = i_n)} \log \frac{\mathbf{Q}'_n(i_n, l)}{\mathbf{Q}_n(i_n, l)}. \end{aligned} \tag{138}$$

Let that we take the first term of (138):

$$\sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1,\ldots,i_n) \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1,l)\ldots\mathbf{Q}_n(i_n,l)}{\mathcal{O}(Y_1=i_1,\ldots,Y_n=i_n)} \log \frac{\mathbf{Q}'_1(i_1,l)}{\mathbf{Q}_1(i_1,l)}$$
$$= \sum_{i_1=1,l=1}^{I_1,K} \sum_{i_2=1,\ldots,i_n=1}^{I_2,\ldots,I_n} \frac{\mathbb{P}(i_1,\ldots,i_n)\mathcal{Q}(i_1,\ldots,i_n,l)}{\mathcal{O}(i_1,\ldots,i_n)} \log \frac{\mathbf{Q}_1(i_1,l)}{\mathbf{Q}_1(i_1,l)}$$
$$= \sum_{i_1=1,l=1}^{I_1,K} \mathbf{Q}'_1(i_1,l) \log \frac{\mathbf{Q}'_1(i_1,l)}{\mathbf{Q}_1(i_1,l)} = D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_1,X} || \mathbb{Q}^{Y_1,X}\right) \tag{139}$$

Now we can note that:

$$\mathbb{E}_{\mathbb{Q}'} D_{\mathrm{KL}}(\mathbb{Q}'^{Y_j|X} || \mathbb{Q}^{Y_j|X})$$
$$= \sum_{l=1}^{K} \left(\sum_{i_1=1,i_2=1,\ldots,i_n=1}^{I_1,I_2,\ldots,I_n} \mathbb{Q}'(i_1,l,i_2,\ldots,i_n) \right) \sum_{i_j=1}^{I_j} \mathbf{Q}'_j(i_j,l) \log \frac{\mathbb{Q}'(i_j,l)}{\mathbb{Q}(i_j,l)}$$
$$= \sum_{l=1}^{K} \sum_{i_1=1}^{I_1} \mathbf{Q}'_1(i_1,l) \sum_{i_j=1}^{I_j} \mathbf{Q}'_j(i_j,l) \log \frac{\mathbf{Q}'_j(i_j,l)}{\mathbf{Q}_j(i_j,l)}. \tag{140}$$

From the update rules (59) we have

$$\sum_{i_1=1,i_j=1}^{I_1,I_j} \mathbf{Q}'_1(i_j,l)\mathbf{Q}'_j(i_j,l)$$
$$= \sum_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1,\ldots,i_n) \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1,l)\ldots\mathbf{Q}_n(i_n,l)}{\mathcal{O}(Y_1=i_1,\ldots,Y_n=i_n)} \tag{141}$$

thus, using (140) for $j = 2,\ldots,n$, we have:

$$\sum\nolimits_{i_1=1,\ldots,i_n=1}^{I_1,\ldots,I_n} \mathbb{P}(i_1,\ldots,i_n) \sum_{l=1}^{K} \frac{\mathbf{Q}_1(i_1,l)\ldots\mathbf{Q}_n(i_n,l)}{\mathcal{O}(Y_1=i_1,\ldots,Y_n=i_n)} \log \frac{\mathbf{Q}'_j(i_j,l)}{\mathbf{Q}_j(i_j,l)}$$
$$= \mathbb{E}_{\mathbb{Q}'} D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_j|X} || \mathbb{Q}^{Y_j|X}\right) \tag{142}$$

using (142) and (139) the Eq. 138 becomes:

$$-A_1 = D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_1,X} || \mathbb{Q}^{Y_1,X}\right) + \sum_j \mathbb{E}_{\mathbb{Q}'} D_{\mathrm{KL}}\left(\mathbb{Q}'^{Y_j|X} || \mathbb{Q}^{Y_j|X}\right). \tag{143}$$

Finally, using (143) and (137) the Eq. 79 becomes (83). The other two formulas are obtained similarly by noticing that optimization of $W_{\mathcal{Q}}\left(\mathbf{Q}'_1\right)$ and the optimization of

$W_{\mathcal{Q}}\left(\mathbf{Q}'_j\right)$ for $j = 2, \ldots, n$ separately yield the same $\mathbf{Q}'_1$ and $\mathbf{Q}'_j$ for $j = 2, \ldots, n$, as those obtained by minimization of W.

References

Bro R, Kiers HAL, Andersson CA (1999) PARAFAC2—part II. Modeling chromatographic data with retention time shifts. J Chemom 13:295–309

Carroll J, Chang 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, Choi S, Plemmons R, Amari S (2007) Non-negative tensor factorization using alpha and beta divergences. In: Proceedings of IEEE international conference on acoustics, speech and signal processing (ICASSP07), vol 3, Honolulu, Hawaii, USA, pp 1393–1396

Cichocki A, Zdunek R, Amari S (2008) Nonnegative matrix and tensor factorization. IEEE Signal Process Mag 25(1):142–145

De Lathauwer L, De Moor B, Vandewalle J (2000) A multilinear singular value decomposition. SIAM J Matrix Anal Appl 21(4):1253–1278

Ding C, He X, Simon HD (2005) On the equivalence of nonnegative matrix factorization and spectral clustering. In: Proceedings of SIAM international conference on data mining, Philadelphia, pp 606–610

Ding C, Li T, Peng W (2008) On the equivalence between non-negative matrix factorization and probabilistic latent semantic indexing. Comput Stat Data Anal 52(8):3913–3927

Donoho D, Stodden V (2004) When does non-negative matrix factorization give a correct decomposition into parts? Adv Neural Inf Process Syst 17

Finesso L, Spreij C (2006) Nonnegative matrix factorization and I-divergence alternative minimization. Linear Algebra Appl 416(2–3):270–286

Friedlander MP, Hatz K (2006) Computing nonnegative tensor factorizations. Technical Report TR-2006-21, University of British Columbia Department of Computer Science

Gaussier E, Goutte C (2005) Relation between PLSA and NMF and implications. In: Proceedings of the 28th annual international ACM SIGIR conference on research and development in information retrieval (SIGIR'05), Salvador, Brazil, pp 601–602

Gonzalez E, Zhang Y (2005) Accelarating the Lee-Seung algorithm for nonnegative matrix factorization. TR-05-02

Harshman RA (1970) Foundations of the PARAFAC procedure: models and conditions for an "explanatory" multi-modal factor analysis. UCLA working papers in phonetics

Hazan T, Polak S, Shashua A (2005) Sparse image coding using a 3D non-negative tensor factorization. In: Tenth IEEE international conference on computer vision, 2005 (ICCV 2005), vol 1, Beijing, China, pp 50–57

Hofmann T (1999) Probabilistic latent semantic indexing. In: Proceedings of the twenty-second annual international SIGIR conference on research and development in information retrieval (SIGIR-99)

Kiers HAL, Berge JMFt, Bro R (1999) PARAFAC2—part I. A direct fitting algorithm for the PARAFAC2 model. J Chemom 13:275–294

Kim Y-D, Choi S (2007) Nonnegative tucker decomposition. In: Proceedings of the IEEE CVPR-2007 workshop on component analysis methods

Kim T-K, Cipolla R (2008) Canonical correlation analysis of video volume tensors for action categorization and detection. In: IEEE Trans Pattern Anal Mach Intell (accepted for publication)

Kim Y-D, Cichocki A, Choi S (2008) Nonnegative Tucker decomposition with alpha-divergence. In: Proceedings of the IEEE international conference on acoustics, speech, and signal processing (ICASSP-2008)

Kolda TG, Bader BW (2009) Tensor Decompositions and applications. SIAM Rev 51:3

Kotsia I, Zafeiriou S, Pitas I (2007) A novel discriminant nonnegative matrix factorization method with application to facial image characterization problems. IEEE Trans Inf Forensics Secur 2(3):588–595

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

Lee DD, Seung HS (1999) Learning the parts of objects by non-negative matrix factorization. Nature 401:788–791

Lee DD, Seung HS (2000) Algorithms for non-negative matrix factorization. In: NIPS, pp 556–562

Lin C-J (2007a) On the convergence of multiplicative update for nonnegative matrix factorization. IEEE Trans Neural Netw (accepted for publication)
Lin C-J (2007b) Projected gradients for nonnegative matrix factorization. Neural Comput (accepted for publication)
Lu H, Plataniotis KN, Venetsanopoulos AN (2008) MPCA: multilinear principal component analysis of tensor objects. IEEE Trans Neural Netw 18(1):18–39
Messer K, Matas J, Kittler JV, Luettin J, Maitre G (1999) XM2VTSDB: the extended M2VTS database. In: AVBPA99, Washington, DC, USA, 22–23 March 1999, pp 72–77
Mørup M, Hansen LK, Arnfred SM (2008) Algorithms for sparse nonnegative tucker decomposition. Neural Comput 20(8):2112–2131
Paatero P, Tapper U (1994) Positive matrix factorization: a non-negative factor model with optimal utilization of error estimates of data values. Environmetrics 111–126
Pascual-Montano A, Carazo JM, Kochi K, Lehmann D, Pascual-Marqui RD (2006) Nonsmooth nonnegative matrix factorization (nsNMF). IEEE Trans Pattern Anal Mach Intell 28(3):403–415
Phillips PJ, Moon H, Rauss PJ, Rizvi S (2000) The FERET evaluation methodology for face recognition algorithms. IEEE Trans Pattern Anal Mach Intell 22(10):1090–1104
Phillips PJ, Wechsler H, Huang J, Rauss P (2009) The FERET database and evaluation procedure for face recognition algorithms. Image Vis Comput 16(5):295–306
Raj RG, Bovik AC (2008) MICA: a multilinear ICA decomposition for natural scene modeling. IEEE Trans Image Process 17(3):259–271
Shashanka MV, Raj B, Smaragdis P (2007) Probabilistic latent variable model for sparse decompositions of non-negative data. Unpublished Draft
Shashanka MV, Raj B, Smaragdis P (2008) Probabilistic latent variable models as non-negative factorizations. Comput Intell Neurosci J
Shashua A, Hazan T (2005) Non-negative tensor factorization with applications to statistics and computer vision. In: International conference of machine learning (ICML)
Shashua A, Zass R, Hazan T (2006) Multi-way clustering using super-symmetric non-negative tensor factorization. In: Proceedings of the European conference on computer vision (ECCV)
Shiryaev AN (1996) Probability, 2nd edn. Springer, Berlin
Sidiropoulos ND, Bro R (2000) On the uniqueness of multilinear decomposition of N-way arrays. J Chemom 37:229–239
Smaragdis P, Raj B (2007) Shift-invariant probabilistic latent component analysis. Mitsubishi Electric Research Laboratories TR2007-009
Smilde A, Bro R, Geladi P (2004) Multi-way analysis: applications in the chemical sciences. Wiley, New York
Sra S, Dhillon IS (2006) Nonnegative Matrix approximation: algorithms and applications. University of Texas at Austin Department of Computer Science Technical Report TR-06-27
Tao D, Li X, Wu X, Maybank SJ (2007) General tensor discriminant analysis and gabor features for gait recognition. IEEE Trans Pattern Anal Mach Intell 10(29):1700–1715
Tucker LR (1966) Some mathematical notes on three-mode factor analysis. Psychometrika 31:279–311
Yan S, Xu D, Yang Q, Zhang L, Tang X, Zhang H-J (2007) Multilinear discriminant analysis for face recognition. IEEE Trans Image Process 1(16):212–220
Zafeiriou S (2009a) Discriminant nonnegative tensor factorization algorithms. IEEE Trans Neural Netw 20(2):217–235
Zafeiriou S (2009b) Algorithms for nonnegative tensor factorization. In: Ferna'ndez SA, de L. Garci'a R, Tao D, Li X (eds) Tensors in image processing and computer vision. Springer, Berlin
Zafeiriou S, Tefas A, Buciu I, Pitas I (2006) Exploitining discriminant information in nonnegative matrix factorization with application to frontal face verification. IEEE Trans Neural Netw 14(8):1063–1073