# Non-Negative Tensor Factorization with Applications to Statistics and Computer Vision

**Amnon Shashua** SHASHUA@CS.HUJI.AC.IL
School of Engineering and Computer Science, The Hebrew University of Jerusalem, Jerusalem 91904, Israel

**Tamir Hazan** TAMIR@CS.HUJI.AC.IL
School of Engineering and Computer Science, The Hebrew University of Jerusalem, Jerusalem 91904, Israel

## Abstract

We derive algorithms for finding a non-negative $n$-dimensional tensor factorization ($n$-NTF) which includes the non-negative matrix factorization (NMF) as a particular case when $n = 2$. We motivate the use of $n$-NTF in three areas of data analysis: (i) connection to latent class models in statistics, (ii) sparse image coding in computer vision, and (iii) model selection problems. We derive a "direct" positive-preserving gradient descent algorithm and an alternating scheme based on repeated multiple rank-1 problems.

## 1. Introduction

Low rank constraints of high dimensional data observations are prevalent in data analysis across numerous scientific disciplines. A factorization of the data into a lower dimensional space introduces a compact basis which if set up appropriately can describe the original data in a concise manner, introduce some immunity to noise and facilitate generalization. Factorization techniques are abundant including Latent Semantic Analysis (Deerwester et al., 1990), probabilistic variants of LSA (Hofmann, 1999), Principal Component Analysis and probabilistic and multinomial versions of PCA (Buntine & Perttu, 2003; Tipping & Bishop, 1999) and more recently non-negative matrix factorization (NMF) (Paatero & Tapper, 1994; Lee & Seung, 1999).

In this paper we address the problem of non-negative factorizations, but instead of two-dimensional data, we handle general $n$-dimensional arrays. In other words, we address the area of non-negative tensor factorizations (NTF) where the NMF is a particular case when $n = 2$. We will motivate the use of $n$-dim NTF in three areas of data analysis: (i) connection to latent class models in statistics, (ii) sparse image coding in computer vision, and (iii) model selection problems.

We will start with the connection between latent class models and NTF, followed by a brief review of what is known about tensor rank factorizations. In Section 4 we will derive our first NTF algorithm based on a direct approach, i.e., a positive preserving (multiplicative) gradient descent over the vectors $\mathbf{u}_i^j$ where $\sum_{j=1}^k \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes ... \otimes \mathbf{u}_n^j$ is a rank-k approximation to the input $n$-way array. In Section 5 we derive an alternative approach based on repeated rank-1 decomposition problems. In Section 6 we apply the NTF to sparse image coding and model selection.

## 2. NTF and Latent Class Models

Consider the joint probability distribution over discrete random variables $X_1, ..., X_n$, where $X_i$ takes values in the set $[d_i] = \{1, ..., d_i\}$. We associate with each entry of the $n$-way array $G_{i_1,i_2,...,i_n}$ a non-negative value $P(X_1 = i_1, ..., X_n = i_n)$ which represents the probability of event $X_1 = i_1, ..., X_n = i_n$.

It is well known that conditional independence constraints among the variables correspond to the zero set of a system of polynomials. For example, a conditional independence statement $A \perp B \mid C$ where $A, B, C$ are pairwise disjoint subsets of $\{X_1, ..., X_n\}$ translates into a set of quadratic polynomials Each polynomial is the determinant of a $2 \times 2$ block of the $n$-way array generated by choosing two distinct elements $a$ and $a'$ in $\prod_{X_i \in A}[d_i]$, two distinct elements $b$ and $b'$ in $\prod_{X_j \in B}[d_j]$ and an element $c$ in $\prod_{X_k \in C}[d_k]$. The determinant is the following expression:

Appearing in *Proceedings of the 22$^{nd}$ International Conference on Machine Learning*, Bonn, Germany, 2005.

$$\begin{aligned} & P(A=a,B=b,C=c)P(A=a',B=b',C=c) \\ - \; & P(A=a',B=b,C=c)P(A=a,B=b',C=c)=0. \end{aligned}$$

The expression on probabilities translates to quadratic expressions on the entries of $G$ by the fact that each probability equals the sum of entries in $G$. For example, If $A \cup B \cup C = \{X_1, ..., X_n\}$ then each probability $P(A=a, B=b, C=c)$ corresponds to a single entry $G_{i_1 i_2 ... i_n}$ where the coordinates of $a, b, c$ have a 1-1 correspondence with $i_1, ..., i_n$.

An algebraically equivalent way of studying the constraints induced by conditional independence statements is by identifying *rank-1 slices* of the tensor $G$. A $k$-way array is said to be of rank-1 if it is described by the outer-product of $k$ vectors $\mathbf{u}_1 \otimes \mathbf{u}_2 \otimes ... \otimes \mathbf{u}_k$. Generally, a statement $A_1 \perp A_2 \perp ... \perp A_l \mid A_{l+1}$ where $A_1, ..., A_{l+1}$ are pairwise disjoint subsets of $\{X_1, ..., X_n\}$ translates to the statement that certain $l$-way slices of $G$ are rank-1 tensors. Consider first the case where $A_1 \cup ... \cup A_{l+1} = \{X_1, ..., X_n\}$. Then, we construct an $(l+1)$-way array whose axes are cartesian products of the $n$ coordinates of $G$ where the first axis is $[d_{i_1}] \times ... \times [d_{i_q}]$ where $X_{i_1}, ..., X_{i_q} \in A_1$, the second axis is $\prod_{X_{i_j} \in A_2} [d_{i_j}]$ and so forth. For every value along the $(l+1)$-axis the remaining $l$-way array (a slice) is a rank-1 tensor. If $A_1 \cup ... \cup A_{l+1} \subset \{X_1, ..., X_n\}$, then $G$ is first "marginalized" (summed over) the remaining variables (those not in the $l+1$ subsets) followed by the construction above.

For example, consider the case of $n = 3$, i.e., we have three random variables $X_1, X_2, X_3$. The statement $X_1 \perp X_2$ translates to the constraint that the matrix resulting from summing over the third coordinate, i.e., $\sum_{i_3} G_{i_1, i_2, i_3}$ is a rank-1 matrix. The statement $X_1 \perp X_2 \mid X_3$ translates to a set of $d_3$ rank-1 constraints: for each value $a \in \{1, ..., d_3\}$ of the third axis $X_3$, the resulting slice $G_{i_1, i_2, a}$ ranging over $i_1, i_2$ is a rank-1 matrix. In probability language,

$$P(X_1, X_2 \mid X_3 = a) = P(X_1 \mid X_3 = a) P(X_2 \mid X_3 = a),$$

is an outer-product of the two vectors $P(X_1 \mid X_3 = a)$ and $P(X_2 \mid X_3 = a)$.

Finally, the statement $X_1 \perp \{X_2, X_3\}$ translates to several rank-1 statements: spread $G$ into a matrix whose first axis is $[d_1]$ and whose second axis is the Cartesian product $[d_2] \times [d_3]$ — the resulting matrix $G_{i_1, i_2 i_3}$ is rank-1. Since $G_{i_1, i_2 i_3}$ is a concatenation of slices $G_{i_1, i_2 a}$ where $X_3 = a$, then each slice must also by rank-1 from which can deduce that $X_1 \perp X_2 \mid X_3$ and likewise since $G_{i_1, a i_3}$ are also slices of $G_{i_1, i_2 i_3}$ then $X_1 \perp X_3 \mid X_2$. Each slice $G_{i_1, i_2 a}$ is of the form $\mathbf{u} \otimes \mathbf{v}_a$ for some fixed vector $\mathbf{u}$ and a vector $\mathbf{v}_a$ that changes from slice to slice. Therefore, the sum of slices (marginalization over $X_3$) is also a rank-1 matrix $\mathbf{u} \otimes (\sum_{i_3} \mathbf{v}_{i_3})$, thus $X_1 \perp X_2$ and likewise $X_1 \perp X_3$.

The introduction of a *latent* (or "hidden") variable $Y$ which takes values in the set $\{1, ..., k\}$ will translate into the fact that slices of $G$ are rank-k tensors, i.e., are described by a sum of $k$ $n$'th fold outer-products. In probability language, the "observed" joint probability $n$-way array $P(X_1, ..., X_n)$ is a marginal of the complete $(n+1)$-way array:

$$P(X_1, ..., X_n) = \sum_{j=1}^{k} P(X_1, ..., X_n, Y = j).$$

As a result, *any conditional independence statement $A_1 \perp A_2 \perp ... \perp A_l \mid A_{l+1}$ over $X_1, ..., X_n$ with a k-graded latent variable $Y$ translates to statements about $l$-way arrays having tensor-rank equal to $k$.*

In probability language, one would say that we have a *mixture* model. The decomposition of the tensor-slice in question into a sum of $k$ rank-1 tensors is equivalent to saying that the probability model is described by a sum of $k$ factors. For example, the basic model of *latent class analysis* is a particular instance (super-symmetric tensors) where the density of an observation $\mathbf{x}^i = (x_1^i, ..., x_n^i)$ is expressed by $f(\mathbf{x}^i) = \sum_{j=1}^k \pi_j f(\mathbf{x}^i; \theta^j)$ where the $j$'th component of the density is given by $f(\mathbf{x}^i; \theta^j) = \prod_{r=1}^n (\theta_r^j)^{x_r^i} (1 - \theta_r^j)^{1 - x_r^i}$.

In probability setting, the method of factorization is the Expectation-Maximization (EM) (Dempster et al., 1977). We will see later in Section 5 the situation in which EM emerges. Generally, recovering the factors is equivalent to a non-negative tensor factorization and that can be studied by algebraic methods — the EM will be shown as a particular instance of the algebraic approach, and *not the best one.*

## 3. What is Known about Tensor Factorizations?

The concept of matrix rank extends quite naturally to higher dimensions: An $n$-valence tensor $G$ of dimensions $[d_1] \times ... \times [d_n]$ is indexed by $n$ indices $i_1, ..., i_n$ with $1 \leq i_j \leq d_j$ is of rank *at most* $k$ if can be expressed as a sum of $k$ rank-1 tensors, i.e. a sum of $n$-fold outer-products: $G = \sum_{j=1}^k \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes ... \otimes \mathbf{u}_n^j$, where $\mathbf{u}_i^j \in R^{d_i}$. The rank of $G$ is the smallest $k$ for which such a decomposition exists.

Despite sharing the same definition, there are a number of striking differences between the cases $n = 2$

(matrix) and $n > 2$ (tensor). While the rank of a matrix can be found in polynomial time using the SVD algorithm, the rank of a tensor is an NP-hard problem. Even worse, with matrices there is a fundamental relationship between rank-1 and rank-k approximations due to the Eckart-Young theorem: the optimal rank-k approximation to a matrix $G$ can be reduced to $k$ successive rank-1 approximation problems to a diminishing residual. This is not true with tensors in general, i.e., repeatedly subtracting the dominant rank-1 tensor is not a converging process, but only under special cases of orthogonally decomposable tensors (see (Zhang & Golub, 2001)).

Another striking difference, this time in favor of tensor ranks, is that unlike matrix factorization, which is generally non-unique for any rank greater than one, a 3-valence tensor decomposition is essentially unique under mild conditions (Kruksal, 1977) and the situation actually improves in higher dimensions $n > 3$ (Sidiropoulos & Bro, 2000). We will address the implications of the uniqueness property in Section 6.

Numerical algorithms for rank-k tensor approximations are abundant. Generalizations of SVD such as orthogonal tensor decompositions (High-Order SVD) have been introduced in (Lathauwer et al., 2000) (and further references therein). Other more general 3-way decompositions were introduced by Harshman (1970) under the name "parallel factor" (PARAFAC) — for a review see (Xianqian & Sidiropoulos, 2001). In computer vision, 3-way tensor decompositions, treating the training images as a 3D cube, have been also proposed (Shashua & Levin, 2001; Vasilescu & Terzopoulos, 2002) with the idea of preserving certain features of the SVD. A recent attempt to perform an NTF was made by (Welling & Weber, 2001) who introduced an iterative update rule, based on flattening the tensor into a matrix representation, but which lacked a convergence proof. As derived next, the key for obtaining a converging update rule is to identify sets of variables with a diagonal Hessian matrix. This is very difficult to isolate when working with matrix representations of tensors and requires working directly with outer-products.

## 4. NTF: the Direct Approach

Given an $n$-way array $G$ we wish to find a non-negative rank-k tensor $\sum_{j=1}^k \mathbf{u}_1^j \otimes \mathbf{u}_2^j \otimes ... \otimes \mathbf{u}_n^j$ described by $nk$ vectors $\mathbf{u}_i^j$. We consider the following least-squares problem:

$$\min_{\mathbf{u}_i^j} \frac{1}{2} \| G - \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j \|_F^2 \qquad \text{subject to} : \mathbf{u}_i^j \geq 0, \quad (1)$$

where $\|A\|_F^2$ is the square Frobenious norm, i.e., the sum of squares of all entries of the tensor elements $A_{i_1,...,i_n}$. The direct approach is to form a positive preserving gradient descent scheme. To that end we begin by deriving the gradient function with respect to $u_{rl}^s$.

Let $< A, B >$ denote the inner-product operation, i.e., $\sum_{i_1,..,i_n} A_{i_1,...,i_n} B_{i_1,...,i_n}$. It is well known that the differential commutes with inner-products, i.e., $d < A, A >= 2 < A, dA >$, hence:

$$\begin{aligned} & \frac{1}{2} d < G - \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j , G - \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j > \\ = & < G - \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j , d \left[ G - \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j \right] > \end{aligned}$$

Taking the differential with respect to $\mathbf{u}_r^s$ and noting that

$$d \left[ G - \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j \right] = - \otimes_{i=1}^{r-1} \mathbf{u}_i^s \otimes d(\mathbf{u}_r^s) \otimes_{i=r+1}^n \mathbf{u}_i^s,$$

the differential becomes:

$$\begin{aligned} df(\mathbf{u}_r^s) &= < \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j \ , \ \otimes_{i=1}^{r-1} \mathbf{u}_i^s \otimes d(\mathbf{u}_r^s) \otimes_{i=r+1}^n \mathbf{u}_i^s > \\ &- < G \ , \ \otimes_{i=1}^{r-1} \mathbf{u}_i^s \otimes d(\mathbf{u}_r^s) \otimes_{i=r+1}^n \mathbf{u}_i^s > \end{aligned}$$

The differential with respect to the $l$'th coordinate $u_{rl}^s$ is:

$$\begin{aligned} df(u_{rl}^s) &= < \sum_{j=1}^k \otimes_{i=1}^n \mathbf{u}_i^j \ , \ \otimes_{i=1}^{r-1} \mathbf{u}_i^s \otimes e_l \otimes_{i=r+1}^n \mathbf{u}_i^s > \\ &- < G \ , \ \otimes_{i=1}^{r-1} \mathbf{u}_i^s \otimes e_l \otimes_{i=r+1}^n \mathbf{u}_i^s > \end{aligned}$$

where $e_l$ is the $l$'th column of the $d_r \times d_r$ identity matrix. Let $S \in [d_1] \times ... \times [d_n]$ represent an $n$-tuple index $\{i_1, ..., i_n\}$. Let $S/i_r$ denote the set $\{i_1, .., i_{r-1}, i_{r+1}, .., i_n\}$ and $S_{i_r \leftarrow l}$ deonte the set of indices $S$ where the index $i_r$ is replaced by the constant $l$. Then, using the identity $< \mathbf{x}_1 \otimes \mathbf{y}_1 \ , \ \mathbf{x}_2 \otimes \mathbf{y}_2 >= (\mathbf{x}_1^\top \mathbf{x}_2)(\mathbf{y}_1^\top \mathbf{y}_2)$ we obtain the partial derivative:

$$\frac{\partial f}{\partial u_{rl}^s} = \sum_{j=1}^k u_{rl}^j \prod_{i \neq r} (\mathbf{u}_i^{j\top} \mathbf{u}_i^s) - \sum_{S/i_r} \left( G_{S_{i_r \leftarrow l}} \prod_{m \neq r} u_{m,i_m}^s \right)$$

Following Lee and Seung (1999) we use a multiplicative update rule by setting the constant $\mu_{rl}^s$ of the gradient descent formula $u_{rl}^s \leftarrow u_{rl}^s - \mu_{rl}^s \frac{\partial f}{\partial u_{rl}^s}$ to be:

$$\mu_{rl}^s = \frac{u_{rl}^s}{\sum_{j=1}^k u_{rl}^j \prod_{i \neq r} (\mathbf{u}_i^{j\top} \mathbf{u}_i^s)}, \qquad (2)$$

thereby obtaining the following update rule:

$$u^s_{rl} \leftarrow \frac{u^s_{rl} \sum_{S/i_r} G_{S_{i_r \leftarrow l}} \prod_{m \neq r} u^s_{m,i_m}}{\sum_{j=1}^k u^j_{rl} \prod_{i \neq r} ({\mathbf{u}^j_i}^\top \mathbf{u}^s_i)} \qquad (3)$$

We will now prove that this update rule reduces the optimization function. The key is that the Hessian matrix with respect to the variables $u^s_{rl}$ is diagonal (and independent of $l$):

$$\frac{\partial^2 f}{\partial u^s_{rl} \partial u^s_{rl}} \quad = \quad \prod_{i \neq r} {\mathbf{u}^s_i}^\top \mathbf{u}^s_i.$$

Moreover, the gradient step $\mu^s_{rl}$ of eqn. 2 is less than the inverse ratio of the Hessian diagonal value:

$$\mu^s_{rl} = \frac{u^s_{rl}}{\sum_{j=1}^k u^j_{rl} \prod_{i \neq r} ({\mathbf{u}^s_i}^\top \mathbf{u}^s_i)} < \frac{u^s_{rl}}{u^s_{rl} \prod_{i \neq r} ({\mathbf{u}^s_i}^\top \mathbf{u}^s_i)}.$$

Finally, we show that the gradient step $\mu^s_{rl} = \mu$ reduces the optimization function.

**Proposition 1** *Let $f(x_1, ..., x_n)$ be a real quadratic function with Hessian of the form $H = cI$ with $c > 0$. Given a point $x = (x^t_1, ..., x^t_n) \in R^n$ and a point $x^{t+1} = x^t - \mu(\bigtriangledown f(x^t))$ with $0 < \mu < \frac{1}{c}$, then $f(x^{t+1}) < f(x^t)$.*

**Proof:** Take the second order Taylor series expansion of $f(x+y)$:

$$f(x+y) = f(x) + \bigtriangledown f(x)^\top y + \frac{1}{2} y^\top H y.$$

Substitute $x = x^t$ and $y = -\mu \bigtriangledown f(x^t)$:

$$\begin{aligned} f(x^t) - f(x^{t+1}) &= \mu \| \bigtriangledown f(x^t) \|^2 - \frac{1}{2}\mu^2 c \| \bigtriangledown f(x^t) \|^2 \\ &= \mu \| \bigtriangledown f(x^t) \|^2 (1 - \frac{1}{2} c\mu) \end{aligned}$$

The result follows since $\mu < \frac{1}{c}$. $\Box$

A similar derivation using the relative entropy error model is possible but is omitted due to lack of space.

## 5. Reduction to Repeated Rank-1 Approximations: $L_2$–EM and EM

We saw in Section 2 that the problem of recovering the $k$ component densities of a latent class model is a special case of finding a rank-k NTF from the joint distribution $n$-way array. In statistics, the component densities are recovered using the EM algorithm. Therefore, an EM-like approach can be considered as an NTF algorithm as well. Formally, we wish to solve the following problem:

$$\min_{W^j, G^j : 1 \leq j \leq k} \| W^j \circ G - G^j \|^2_F$$
$$\text{s.t. } G^j \geq 0, \; rank(G^j) = 1, \; \sum_j W^j = \mathbf{1},$$

where $A \circ B$ stands for the element-wise (Hadamard) product of the arrays, $\mathbf{1}$ is the $n$-array of 1s, and $\sum_j G^j$ is the sought-after rank-k decomposition. Note that for every choice of $W^j$ which sum-up to the unit tensor $\mathbf{1}$ we have: $\sum_j W^j \circ G = G$, thus the requirement $G = \sum_j G^j$ is implied by the conditions above. We will be alternating between optimizing one set of variables while holding the other set constant, thus breaking down the problem into alteration between two (convex) optimization problems: (i) given current estimate of $W^j$, solve for $G^j$ by finding the closest rank-1 fit to $W^j \circ G$, and (ii) given current estimates of $G^j$, solve for $W^j$.

The advantage of this approach is that it reduces the rank-k approximation problem to multiple and repeated rank-1 approximations. The advantage is twofold: on one hand a rank-1 approximation can be achieved by a straightforward extension of the power method for finding the leading eigenvector of a matrix, but moreover, the rank-1 approximation carries with it properties that are difficult to guarantee when seeking a rank-k approximation directly. For example, if $G$ is super-symmetric (i.e., the rank-1 factors are $n$-fold symmetric) then the rank-k approximation described in the previous section will not necessarily find a super-symmetric decomposition, i.e., where $\mathbf{u}^j_1 = ... = \mathbf{u}^j_n$, but a rank-1 fit to a super-symmetric tensor is guaranteed to have a symmetric form (cf. Catral et al. (2004),Kofidis and Regalia (2002)).

Given $W^j \geq 0$, the optimal $G^j$ can be found by fitting a rank-1 tensor to $W^j \circ G$. A least-squares fit of $\mathbf{u}_1 \otimes ... \otimes \mathbf{u}_n$ to a given tensor $H$ can be achieved by employing the following "power method" scheme (see (Zhang & Golub, 2001)) summarized in Fig. 1. The update process preserves positivity, so if $W^j \geq 0$ and $G \geq 0$ then the resulting rank-1 fit is also non-negative.

We next consider the problem of finding the optimal $W^1, ..., W^k$ satisfying the admissibility constraints $\sum_j W^j = \mathbf{1}$ and $W^j \geq 0$ given that we know the values of $G^1, ..., G^k$. Let $S$ as before stand for the index $i_1, .., i_n$ into the $n$-way arrays. Let $\mathbf{b}_S = (1/G_S)(G^1_S, ..., G^k_S)$ and $\mathbf{q}_S = (W^1_S, ..., W^k_S)$, then our problem is to find the $k$-dimensional vectors $\mathbf{q}_S$ for all $S$ ranging in $[d_1] \times ...[d_n]$ which satisfy the following

```
Input: The $n$-way array $G$ and the current estimate of $W^j$.
Output: The closest least-squares rank=1 approximation $G^j$ to $W^j \circ G$.

1. Let $H = W^j \circ G$.
2. Initialize the vectors $\mathbf{u}_1^{(0)}, ..., \mathbf{u}_n^{(0)}$, where $\mathbf{u}_r^{(0)} \in R^{d_r}$, to random non-negative values.
3. for $t = 0, 1, 2, .., T$
    (a) $u_{r,i_r}^{(t+1)} = \sum_{S/i_r} H_{S/i_r} \prod_{m=1}^{r-1} u_{m,i_m}^{(t+1)} \prod_{m=r+1}^{n} u_{m,i_m}^{(t)}$, where $r = 1, ..., n$ and $i_r = 1, ..., d_r$.
    (b) replace $\mathbf{u}_r^{(t+1)} \leftarrow \mathbf{u}_r^{(t+1)} / \|\mathbf{u}_r^{(t+1)}\|$.
4. $G^j = \delta \mathbf{u}_1^{(T)} \otimes .... \otimes \mathbf{u}_n^{(T)}$, where $\delta =< H, \otimes_{i=1}^{n} \mathbf{u}_i^{(T)} >$.
```

*Figure 1.* $L_2$ Alternating Scheme: the power method for finding the closest rank=1 tensor $G^j$ to the given tensor $W^j \circ G$. The symbol $S$ represents an in index $i_1, ..., i_n$ and $S/i_r$ the index $i_1, ..., i_{r-1}, i_{r+1}, ..., i_n$.

optimization criteria:

$$\min_{\mathbf{q}_S} \sum_{S \in [d_1] \times ... \times [d_n]} \|\mathbf{q}_S - \mathbf{b}_S\|_2^2 \quad \text{s.t. } \mathbf{q}_S \geq 0, \ \mathbf{q}_S^\top \mathbf{1} = 1.$$

Since this problem is solved for each index $S$ separately (there is no coupling between $\mathbf{q}_S$ and $\mathbf{q}_{S'}$), we can omit the reference to the index $S$ and focus on solving the following problem:

$$\mathbf{q}^* = \text{argmin}_{\mathbf{q}} \|\mathbf{q} - \mathbf{b}\|_2^2 \quad \text{s.t. } \mathbf{q} \geq 0, \ \mathbf{q}^\top \mathbf{1} = 1 \qquad (4)$$

The optimal solution $\mathbf{q}^*$ can be found by employing an iterative scheme alternating between the following two partial optimizations: Let $\mathbf{q}_+^{(0)} = \mathbf{b}$, then for $t = 0, 1, 2, ....$ we define:

$$\mathbf{q}^{(t+1)} = \text{argmin}_{\mathbf{x}} \|\mathbf{x} - \mathbf{q}_+^{(t)}\|^2 \quad \text{s.t. } \mathbf{x}^\top \mathbf{1} = 1, \quad (5)$$

$$\mathbf{q}_+^{(t+1)} = \text{argmin}_{\mathbf{x}} \|\mathbf{x} - \mathbf{q}^{(t+1)}\|^2 \quad \text{s.t. } \mathbf{x} \geq 0. \quad (6)$$

See Fig. 3 for a sketch of the alternating steps. Following some algebra, the *optimal* solution $\mathbf{q}^*$ for eqn. 4 is obtained by the iterative scheme described in Fig. 2. We omit the convergence and global optimality proofs due to lack of space.

To summarize, the alternating scheme, referred to as $L_2$–EM, for updating the estimates $G^1, ..., G^k$ is as follows:

1. Let $W^{1^{(0)}}, ..., W^{k^{(0)}}$ be assigned random values in the range $[0, 1]$ and normalized such that $\sum_j W^{j^{(0)}} = \mathbf{1}$.
2. Let $t = 0, 1, 2, ...$
3. Assign $G^{r^{(t)}} \leftarrow rank1(W^{r^{(t)}} \circ G)$, $r = 1, ..., k$, using the power method presented in Fig. 1.

```
Input: The $n$-way array $G$ and the current rank1 factors $G^1, ..., G^k$.
Output: The updated estimate of the $n$-way arrays $W^1, ..., W^k$.

1. for $S = i_1, ..., i_n \in [d_1] \times .... \times [d_n]$.
    (a) Let $\mathbf{q}_+^{(0)} = (1/G_S)(G_S^1, ..., G_S^k)$.
    (b) for $t = 0, 1, 2, 3, ...$
        i. $q_j^{(t+1)} = q_{+_j}^{(t)} + \frac{1}{k}(1 - \sum_{l=1}^{k} q_{+_l}^{(t)})$, $j = 1, ..., k$.
        ii. $\mathbf{q}_+^{(t+1)} = th_{\geq 0}(\mathbf{q}^{(t+1)})$.
        iii. repeat until $\mathbf{q}^{(t+1)} = \mathbf{q}_+^{(t+1)}$.
    (c) Let $W_S^j = q_j^{(t+1)}$, $j = 1, ..., k$.
```

*Figure 2.* The iterative scheme for updating $W^1, ..., W^k$ given the current estimate of the factors $G^1, ..., G^k$ and the input tensor $G$.

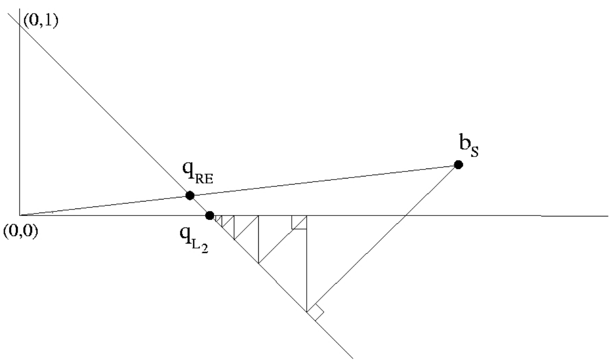

*Figure 3. A sketch of the convergence pattern of the $L_2$ update of the auxiliary tensors $W^1, ..., W^k$. The pattern proceeds by successive projections onto the hyperplane $\mathbf{x}^\top \mathbf{1} - 1 = 0$ followed by projection to the non-negative orthant $\mathbf{x} \geq 0$. The relative entropy update resulting in $\mathbf{q}_{re}$ is simply a scaling of $\mathbf{b}_S$ thus non-vanishing entries of $\mathbf{b}_S$ cannot map to vanishing entries of $\mathbf{q}_{re}$. On the other hand, non-vanishing entries of $\mathbf{b}_S$ can map to vanishing entries of $\mathbf{q}_{L2}$.*

4. Assign values to $W^{1^{(t+1)}}, ..., W^{k^{(t+1)}}$ using the iterative scheme presented in Fig. 2 with the estimates of $G^{r^{(t)}}$, $r = 1, ..., k$.
5. Repeat until convergence.

It is worthwhile noting that if we replace the $L_2$ error model with the relative entropy $D(A \parallel B) = \sum_S \left[ A_S \log \frac{A_S}{B_S} - A_S + B_S \right]$ and go through the algebra we obtain the EM algorithm. Specifically, given the current estimates $G^1, .., G^k$ the problem of finding the optimal (under relative entropy) $W^r$ is convex and after some algebra can be shown to be equal to:

$$W^r = \frac{G^r}{\sum_{j=1}^{k} G^j}. \qquad (7)$$

This is a non-iterative update rule which involves only scaling — see Fig. 3 for a sketch comparing this with the $L_2$ result. The entries of $W^r$ represent $P(y_i = r \mid \mathbf{x}_i, \theta)$ indicating the probability of the co-occurence value to arise from the $r$'th factor. The update formula of eqn. (7) is the Bayes rule:

$$P(y_i = r \mid \mathbf{x}_i, \theta') = \frac{P(\mathbf{x}_i \mid y_i = r, \theta')P(y_i = r \mid \theta')}{P(\mathbf{x}_i \mid \theta')},$$

where $\theta'$ are the parameters (the $\mathbf{u}_i^j$ making up the $G^j$, $j = 1, ..., k$) of the previous iteration.

Given the current estimates of $W^1, ..., W^k$ the optimal $G^r$ is the rank-1 fit to $W^r \circ G$. The rank-1 fit of $\mathbf{u}_1 \otimes ... \otimes \mathbf{u}_n$ to a tensor $H$ under relative entropy can be shown to be equal to:

$$u_{r,i_r} = \sum_{i_1,...,i_{r-1},i_{r+1},...,i_n} H_{i_1,...,i_n}, \qquad (8)$$

normalized by the sum of all entries of $H$ (the $L_1$ norm of $\mathbf{u}_r$ is 1). The maximization step of EM is over the following function:

$$\max_{\theta} \sum_{i=1}^{l} \sum_{j=1}^{k} P(y_i = j \mid \mathbf{x}_i, \theta^{(t)}) \log P(\mathbf{x}_i, y_i = j \mid \theta),$$

where $\mathbf{x}_i$ are the i.i.d. samples. Given that each sample appears multiple times in order to create a co-occurence array $G$, the problem is equivalent to:

$$\max_{\mathbf{u}_i^j \geq 0} \sum_{S} \sum_{j=1}^{k} w_S^j G_S \log u_{1,i_1}^j \cdots u_{n,i_n}^j,$$

subject to $\|\mathbf{u}_i^j\|_1 = 1$ and where $S$ runs over the indices $\{i_1, ..., i_n\}$. Taking the derivatives with respect to the vectors $\mathbf{u}_i^j$ and setting them to zero will provide the update rule of eqn. (8) — thereby establishing the equivalence of the two update formulas eqns. 7 and 8 with EM.

In the remainder of this section we wish to analyze the difference between the solutions the two schemes, the $L_2$–EM and EM, can provide. Consider the rank-1 fit step: a rank-1 $\mathbf{u}\mathbf{v}^\top$ fit to a matrix $A$ would be $\mathbf{u} = A\mathbf{1}$ and $\mathbf{v} = A^\top\mathbf{1}$, i.e., the "closest" rank-1 matrix to $A$ is $A\mathbf{1}\mathbf{1}^\top A$. In contrast, an $L_2$ fit would generate $\mathbf{u}$ as the leading eigenvector of $A$ and $\mathbf{v}$ as the leading eigenvector of $A^\top$. Clearly, if $A$ is a random matrix then both approximations will coincide since the vector $\mathbf{1}$ is the leading eigenvector of a random matrix. For non-random matrices, the $L_2$ rank-1 fit tends to be more sparse then its relative entropy counterpart — as stated in the following claim:

**Proposition 2** *Consider a non-negative matrix $A$ and let $\mathbf{u}_{L2}\mathbf{v}_{L2}^\top$ be the $L_2$ rank-1 fit to $A$ and $\mathbf{u}_{RE}\mathbf{v}_{RE}^\top$ be the relative entropy rank-1 fit to $A$. Then,*

$$\|\mathbf{u}_{L2}\|_0^0 \leq \|\mathbf{u}_{RE}\|_0^0, \quad \text{and} \quad \|\mathbf{v}_{L2}\|_0^0 \leq \|\mathbf{v}_{RE}\|_0^0,$$

*where $\|\mathbf{u}\|_0^0 = \#\{i : u_i \neq 0\}$ the zero-norm of $\mathbf{u}$.*

We omit the proof due to lack of space. The proposition holds (under mild conditions) for higher order tensors as well. A similar situation holds for the update of the $W^j$ tensors as can be seen from the sketch of Fig. 3. The implication of this result is that we should expect a higher sensitivity to noise with the relative entropy scheme compared to the $L_2$ norm counterpart — this would be explored empirically in the next section.

## 6. Experiments

We will begin with exploring the differences between NMF and NTF. Any $n$-dimensional problem can be represented in two dimensional form by concatenating dimensions. Thus for example, the problem of finding a non-negative low rank decomposition of a set of images is a 3-NTF (the images forming the slices of a 3D cube) but can also be represented as an NMF problem by vectorizing the images (images forming columns of a matrix). There are two reasons why a matrix representation of a collection of images would not be appropriate: (i) spatial redundancy (pixels, not necessarily neighboring, having similar values) is lost in the vectorization thus we would expect a less efficient factorization (a point made and demonstrated in Shashua and Levin (2001)), and (ii) an NMF decomposition is not unique therefore even if there exists a generative model (of local parts) the NMF would not necessarily move in that direction — a point made by (Donoho & Stodden, 2003) and verified empirically by others (Chu et al., 2004). For example, invariant parts on the image set would tend to form ghosts in all the factors and contaminate the sparsity effect. As mentioned in Section 3, an NTF is almost always unique thus we would expect the NTF scheme to move towards the generative model, and specifically not be influenced by invariant parts.

Following (Donoho & Stodden, 2003) we built the Swimmer image set of 256 images of dimensions 32 × 32. 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. Fig. 4, second row, shows 6 (out of 17) factors using an NMF representation (running the Lee-Seung algorithm). The torso being an invariant part, as it appears in the same position through the entire set, appears as a "ghost" in all the factors. On the other hand, the NTF factors (third row) resolve correctly all the 17 parts. The number of rank-1 factors is 50 (since the diagonal limbs are not rank-1 parts). The rank-1 matrices corresponding to the limbs are superimposed in the display in Fig. 4 for purposes of clarity.

The 4th row of Fig. 4 shows some of the NMF factors generated from a set of 2429, 19×19, images faces from the MIT CBCL database. One can clearly see ghost structures and the part decomposition is complicated (an observation supported by empirical studies done by other groups such as on Iris image sets in (Chu et al., 2004)). The NTF factors (rank-1 matrices), shown in the 5th row, have a sharper decomposition into sparse components. The 6th row shows an overlay of rank-1 factors whose energy are localized in the same image region — we have done that for display purposes. One can clearly see the parts (which now correspond to higher rank matrices) corresponding to eyes, cheeks, shoulders, etc.

Since NTF preserves the image spatial dimension one would expect a higher efficiency rate (in terms of compression) compared to an NMF coding. Indeed, the reconstruction quality of the original images from the factors roughly agree when the compression ratio between NMF to NTF is 1 to 10, i.e., a reconstruction with 50 NTF factors (each factor is represented by 38 numbers) is comparable to the performance with 50 NMF factors (each factor is represented by $19^2 = 362$ numbers). This reflects on efficiency as the number of rank-1 components required to represent a set of images would be significantly smaller with an NTF decomposition.

To test the implication of Proposition 2 to noise sensitivity we have taken one of the Swimmer pictures and created a 3D tensor by taking 20 copies of the picture as slices of a 3D cube where to each copy a random pattern (noise) was superimposed. We then looked for a rank-2 decomposition (in fact there are 7 factors but we looked for the dominant two factors). The factors found by the $L_2$–EM scheme were indeed sparser and with a much closer fit to the original factors than those generated by the EM scheme.

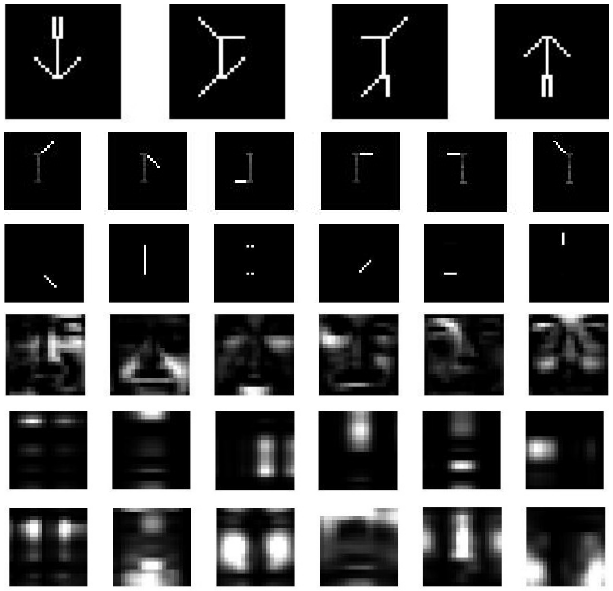

*Figure 4. Comparing the factors generated by NMF (second row) and NTF (third row) from a set of 256 images of the Swimmer library (sample in top row). The NMF factors contains ghosts of invariant parts (the torso) which contaminate the sparse representation. 4th row: leading NMF factors of CBCL face dataset, compared to leading NTF factors in the 5th row. 6th row: summed factors of NTF located in the same region (resulting in higher rank factors) — see text for explanation.*

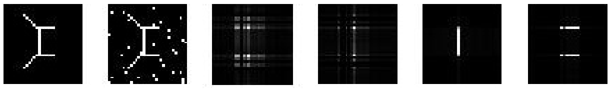

*Figure 5. Sensitivity of the alternating schemes to noise. Left-to-right: original image, image superimposed with a random pattern, the leading pair of rank-1 factors generated by the EM scheme, and the leading pair generated by the $L_2$–EM scheme.*

We next show some preliminary results of using NTF for model selection. A super-symmetric $n$-way array corresponds to a model selection problem where a model is determined by $q < n$ points. We consider two examples. In Fig. 6 we have two views of a 3D scene with two moving bodies: the background motion generated by the motion of the camera and the motion of the vehicles. Assuming an orthographic projection model, a matching pair with coordinates $(x, y)$ and $(x', y')$ satisfy a linear constraint $\alpha_1 x + \alpha_2 y + \alpha_3 x' + \alpha_4 y' + \alpha_5 = 0$ with fixed coefficients (Ullman & Basri, 1991) — therefore, a minimum of 5 matching points are necessary for verifying whether they come from the same body. The probability of a $n$-tuple ($n > 4$) of matching points to arise from the same body is represented by $\exp^{-\lambda}$ where $\lambda$ is the least significant eigenvalue (the residual) of the $5 \times 5$ measurement matrix $A^\top A$ where the rows of $A$ are the vectors $(x_i, y_i, x'_i, y'_i, 1)$, $i = 1, ..., n$. We have chosen $n = 7$ and generated a 7-NTF with 100 non-vanishing entries (i.e., we sampled 100 7-tuples) sampled over 30 matching points across the two views. We performed a super-symmetric weighted NTF (where the zero weights correspond to the vanishing entries of the tensor). Due to the super-symmetry, each factor (a

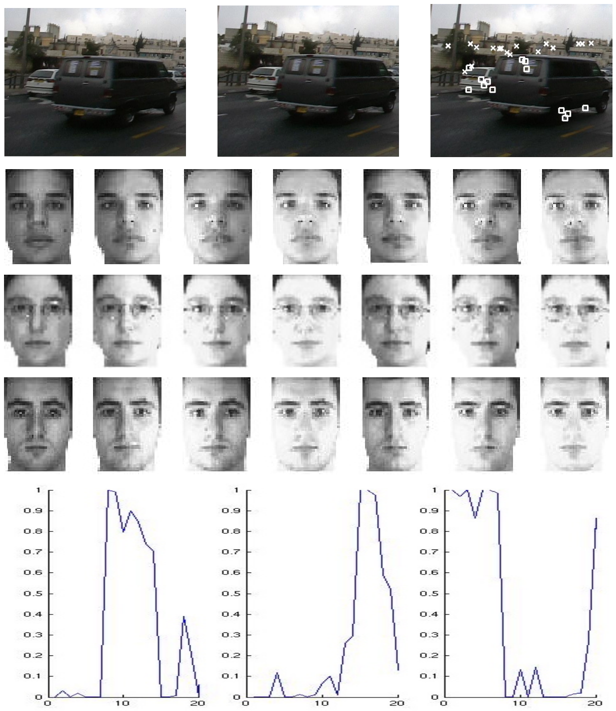

*Figure 6. Performing model selection NTF. First row illustrates affine multi-body segmentation and rows* $2-5$ *illustrate recognition under varying illumination. See text for details.*

7-way array) is represented by a single vector of 30 entries. Each entry corresponds to the probability of the corresponding point to belong to the object represented by the factor — this comes directly from the latent class model connection with NTF. The segmentation result is shown in Fig. 6 — we obtain a perfect segmentation with a relatively small number of samples.

Rows $2-4$ of Fig. 6 shows three persons under varying illumination conditions. Using the result that matte surfaces under changing illumination live in a 3D subspace (Shashua, 1997) we create a super-symmetric 4-NTF where each entry corresponds to the probability that 4 pictures (sampled from the 21 pictures) belong to the same person. The first three factors of the NTF correspond to the probability that a picture belongs to the person represented by the factor. The factors are shown in the 5th row where one can see an accurate clustering of the pictures according to the three different persons.

The details of performing a super-symmetric NTF and how to incorporate the weights are relatively straightforward but are left to future publication due to lack of space.

## References

Buntine, W., & Perttu, S. (2003). Is multinomial pca multi-faceted clustering or dimensionality reduction. *Proc. 9th Int. Workshop on Artificial Intelligence and Statistics* (pp. 30–307).

Catral, M., Han, L., Neumann, M., & Plemmons, R. (2004). On reduced rank for symmetric nonnegative matrices. *Linear Algebra and its Applicatoins*, *393*, 107–126.

Chu, M., Diele, F., Plemmons, R., & Ragni, S. (2004). Optimality, computation and interpretation of nonnegative matrix factorizations. *SIAM Journal on Matrix Analysis.*

Deerwester, A., Dumanis, S., Furnas, G., T.K., L., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society for Information Science*, *41*.

Dempster, A., Laird, N., & Rubin, D. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Stat. Soc., series B*, *39*, 1–38.

Donoho, D., & Stodden, V. (2003). When does non-negative matrix factorization give a correct decomposition into parts. *Proceedings of the conference on Neural Information Processing Systems (NIPS).*

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

Hofmann, T. (1999). Probabilistic latent semantic analysis. *Proc. of Uncertainty in Artificial Intelligence, UAI'99.* Stockholm.

Kofidis, E., & Regalia, P. (2002). On the best rank-1 approxiamtion of higher order supersymmetric tensors. *Matrix Analysis and Applications*, *23*, 863–884.

Kruksal, J. (1977). Three way arrays: rank and uniqueness of trilinear decomposition, with application to arithmetic complexity and statistics. *Linear Algebra and its Applications*, *18*, 95–138.

Lathauwer, L. D., Moor, B. D., & Vandewalle, J. (2000). A multilinear singular value decomposition. *Matrix Analysis and Application*, *21*, 1253–1278.

Lee, D., & Seung, H. (1999). Learning the parts of objects by non-negative matrix factorization. *Nature*, *401*, 788–791.

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

Shashua, A. (1997). On photometric issues in 3D visual recognition from a single 2D image. *International Journal of Computer Vision*, *21*, 99–122.

Shashua, A., & Levin, A. (2001). Linear image coding for regression and classification using the tensor-rank principle. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition.* Hawaii.

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

Tipping, M., & Bishop, C. (1999). Probabilistic principal component analysis. *Journal of the Royal Statistical Society, Series B,21(3):611-622.*

Ullman, S., & Basri, R. (1991). Recognition by linear combination of models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *13*, 992–1006.

Vasilescu, M., & Terzopoulos, D. (2002). Multilinear analysis of image ensembles: Tensorfaces. *Proceedings of the European Conference on Computer Vision* (pp. 447–460).

Welling, M., & Weber, M. (2001). Positive tensor factorization. *Pattern Recognition Letters*, *22*, 1255–1261.

Xianqian, L., & Sidiropoulos, N. (2001). Cramer-rao lower boubds for low-rank decomposition of multidimensional arrays. *IEEE Transactions on Signal Processing*, *49*.

Zhang, T., & Golub, G. (2001). Rank-one approximation to high order tensors. *Matrix Analysis and Applications*, *23*, 534–550.