

Nonnegative Factorization of Diffusion Tensor Images and Its Applications

Yuchen Xie, Jeffrey Ho, and Baba C. Vemuri*

Department of CISE, University of Florida, Gainesville FL, 32611, USA
{yxie, jho, vemuri}@cise.ufl.edu

Abstract. This paper proposes a novel method for computing linear basis images from tensor-valued image data. As a generalization of the nonnegative matrix factorization, the proposed method aims to approximate a collection of diffusion tensor images using nonnegative linear combinations of basis tensor images. An efficient iterative optimization algorithm is proposed to solve this factorization problem. We present two applications: the DTI segmentation problem and a novel approach to discover informative and common parts in a collection of diffusion tensor images. The proposed method has been validated using both synthetic and real data, and experimental results have shown that it offers a competitive alternative to current state-of-the-arts in terms of accuracy and efficiency.

1 Introduction

In this paper, we introduce the novel notion of nonnegative factorization of tensor fields and its application to the segmentation of diffusion tensor images (DTI). Tensor images (or tensor fields) abound in medical imaging applications, with well-known examples such as DTIs, conductivity tensor images (CTI) and elasticity tensors in Elastography. There is an increasing demand for a principled and versatile method that can automatically extract important features from single as well as multiple tensor images, and this paper takes a step forward in this direction by investigating the general factorization problem for tensor fields. Specifically, we formulate the factorization as an optimization problem, and as the main technical contribution, we present an iterative algorithm that can efficiently and reliably optimize the objective function and compute good-quality factorizations.

Matrix factorization in the form of factoring a data matrix \mathbf{V} into a product of the basis matrix \mathbf{W} and the coefficient matrix \mathbf{H} appears frequently in computer vision and image processing. Examples include the principal component analysis (PCA) and the Tomasi-Kanade factorization from multi-view 3D structure recovery. Algebraically, the factorization attempts to “discover” the linear subspace spanned by the columns of \mathbf{W} that can be used to approximate the data given as the columns of \mathbf{V} . Instead of the full subspace, the linear model can be further constrained by requiring that the data vectors belong to (or can be approximated by) the cone generated by the columns of \mathbf{W} . Since

* This research is in part supported by the NIH grant NS066340 to BCV, the NSF grant IIS 0916001 to JH and BCV and the University of Florida Alumni Fellowship to YX.

a point in a cone can be written as a nonnegative linear combination of the cone’s generators, this gives rise to the nonnegative matrix factorization (NMF) first introduced in [7]. Specifically, using images as data, the matrices appeared in an NMF are all required to have nonnegative components. While the unconstrained matrix factorization admits a comparatively straightforward solution using linear algebra, nonnegative matrix factorization is generally more difficult to solve as its objective function is non-convex and there exists no known linear algebraic methods that can compute its solution in closed form. Nevertheless, the optimization is not particularly involved as the nonnegative constraint can be easily enforced, and NMF is almost always solved using an alternating sequence of nonnegative linear least squares with variables in \mathbf{W} and \mathbf{H} separately. More importantly, the nonnegativity constraint has been argued in the original paper [7] to be more capable, compared with the unconstrained factorization, of capturing features that are common among the data. In particular, the columns of the basis matrix \mathbf{W} can be considered as the discovered *common parts* among the input data, and the coefficient matrix \mathbf{H} provides the *weights* for reconstructing each input data using these parts. The power and versatility of NMF have been demonstrated to various degrees by a wide range of successful applications that include data clustering [2], parts discovery [9] and MR image analysis [6].

The proposed nonnegative factorization of tensor-valued images is an extension of the nonnegative matrix factorization method above. In our context, components in the matrices \mathbf{V} and \mathbf{W} are represented as symmetric positive semi-definite (PSD) tensors of rank two and \mathbf{H} is the coefficient matrix with nonnegative real components. In particular, the nonnegativity constraint in our more general setting has now acquired two different realizations: the nonnegative constraint on the components of \mathbf{H} as in the original NMF and the generalized nonnegative constraint on the components of \mathbf{W} that they belong to the space of $\mathbf{PSD}(n)$ of $n \times n$ symmetric positive semi-definite matrices. Geometrically, this corresponds to replacing the nonnegative real line (intensity values) with the space $\mathbf{PSD}(n)$ (diffusion tensors), and it is their respective cone structures that permit us to formulate nonnegative factorization using these spaces. While our extension is easy to understand conceptually, the resulting optimization problem in our case is considerably more difficult to solve because of the generalized nonnegative constraint on \mathbf{W} and the dimension of $\mathbf{PSD}(3)$. The former requires generalized matrix inequalities to enforce the constraint [1] and the latter introduces a large number of variables in the objective function. Therefore, a major portion of this paper is devoted to an optimization algorithm that can reliably and efficiently compute the factorization.

Having overcome this computational hurdle, we will next show that the proposed tensor images factorization method can be successfully applied to segmentation problems for single and multiple DTI images. For a single image, the data matrix \mathbf{V} is an $1 \times n$ array of PSD tensors, and a direct clustering on the coefficient matrix \mathbf{H} gives the segmentation. For multiple tensor images, the basis matrix \mathbf{W} gives as before a “part-decomposition” of the collection, and the common parts given as the columns of \mathbf{W} can usually be realized as tensor fields with local support. In the current medical imaging literature, algorithms that segment DTI images can be broadly classified into two categories, the level-set based methods (e.g., [16, 8]) and the methods based on combinatorial optimization such as graph cuts [17]. In terms of its underlying motivation

and numerics, our factorization-based method offers a completely different approach to the segmentation problem. We have validated the proposed method using both synthetic and real tensor images. Preliminary segmentation results from single and multiple DTI images have indicated that the proposed factorization-based approach is a viable and competitive alternative method for segmenting tensor images.

2 Preliminaries

Given an $n \times m$ non-negative matrix \mathbf{V} , each of whose column \mathbf{v}_i represents an input image, non-negative matrix factorization (NMF) [7] attempts to factor it into two matrices with non-negative components, $\mathbf{V} \approx \mathbf{W}\mathbf{H}$, where \mathbf{W} denotes the $n \times r$ basis matrix and \mathbf{H} denotes the $r \times m$ coefficient matrix. According to the geometric interpretation elucidated in [3], NMF determines a cone $\Sigma_{\mathbf{W}} = \{\mathbf{x} | \mathbf{x} = \sum_{j=1}^r h_j \mathbf{w}_j, h_j \geq 0\}$ that approximates the input images with the basis elements (or generators of $\Sigma_{\mathbf{W}}$) \mathbf{w}_j given by the j -th column of \mathbf{W} .

For tensor-valued images, there is a $d \times d$ symmetric, positive semi-definite matrix associated to each pixel (or voxel). $d = 3$ for diffusion tensor images. The mathematics for nonnegative factorization of tensor images is straightforward as it only requires a cone structure that is provided by $\mathbf{PSD}(d)$, the space of $d \times d$ symmetric positive semi-definite matrices. A collection of m tensor images of size n can be arranged into a

block matrix $\mathbf{V} = \begin{pmatrix} V_{11} & \cdots & V_{1m} \\ \vdots & \ddots & \vdots \\ V_{n1} & \cdots & V_{nm} \end{pmatrix}$, where $V_{ki} \in \mathbf{PSD}(d)$, $k = 1, \dots, n$ and $i =$

$1, \dots, m$. Each of the m columns represents one tensor image. As before, nonnegative factorization for tensor images attempts to factor \mathbf{V} into a product of the basis matrix \mathbf{W} the coefficient matrix \mathbf{H} whose elements are non-negative real numbers:

$$\mathbf{V} \approx \mathbf{W} * \mathbf{H} = \begin{pmatrix} W_{11} & \cdots & W_{1r} \\ \vdots & \ddots & \vdots \\ W_{n1} & \cdots & W_{nr} \end{pmatrix} * \begin{pmatrix} h_{11} & \cdots & h_{1m} \\ \vdots & \ddots & \vdots \\ h_{r1} & \cdots & h_{rm} \end{pmatrix}, \quad (1)$$

where the blocks W_{ij} in \mathbf{W} are matrices in $\mathbf{PSD}(d)$, and the blockwise product $*$ is defined as

$$\mathbf{W} * \mathbf{H} = \begin{pmatrix} \sum_{j=1}^r W_{1j} h_{j1} & \cdots & \sum_{j=1}^r W_{1j} h_{jm} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^r W_{nj} h_{j1} & \cdots & \sum_{j=1}^r W_{nj} h_{jm} \end{pmatrix}. \quad (2)$$

r in the above equation is the number of basis elements (generators) used in the factorization, and it is clear that our nonnegative factorization reduces to the usual NMF when $d = 1$. We remark that the nonnegative factorization can be considered as a generative model for the collection of input tensor images as each input tensor image is approximated by a non-negative linear combination of r columns of \mathbf{W} , each of which can be considered as a tensor image. To determine \mathbf{W} , \mathbf{H} from the data matrix \mathbf{V} , we

formulate a constrained optimization problem that minimizes the cost function

$$E(\mathbf{W}, \mathbf{H}) = \frac{1}{2} \sum_{i=1}^m \sum_{k=1}^n \|V_{ki} - \sum_{j=1}^r W_{kj} h_{ji}\|_F^2 \quad (3)$$

with the constraints $W_{kj} \succeq 0$ and $h_{ji} \geq 0$, $i = 1, \dots, m$, $j = 1, \dots, r$ and $k = 1, \dots, n$. \succeq denotes matrix inequality and $\|\cdot\|_F$ denotes the Frobenius norm.

3 Algorithm and implementation details

In this section, we present an efficient algorithm that solves the constrained optimization defined above. While the objective function $E(\mathbf{W}, \mathbf{H})$ is not convex, it is convex with respect to the two block variables \mathbf{W} and \mathbf{H} . A common approach to solve this type of constrained optimization problem is the block coordinate descent method [10], and in particular, we can alternatively fix one block variable and improve the other as shown in Algorithm 1. Grippo et al [5] have shown that every limit point of the sequence

Algorithm 1 Alternating Non-negative Factorization

Initialize $\mathbf{H}^1 \geq 0$.

For $t = 1, 2, \dots$

- $\mathbf{W}^{t+1} = \underset{\mathbf{W}}{\operatorname{argmin}} E(\mathbf{W}, \mathbf{H}^t)$, s.t. $W_{kj} \succeq 0, \forall k, j$.
 - $\mathbf{H}^{t+1} = \underset{\mathbf{H}}{\operatorname{argmin}} E(\mathbf{W}^{t+1}, \mathbf{H})$, s.t. $\mathbf{H} \geq 0$.
-

$\{\mathbf{W}^t, \mathbf{H}^t\}$ generated by Algorithm 1 is a stationary point of the optimization problem defined in (3). Since both sub-problems are convex, our algorithm can easily be shown to be provably convergent. The details for solving the two sub-problems efficiently will be discussed below.

3.1 Optimization with respect to the basis matrix \mathbf{W}

When \mathbf{H} is fixed, the optimization problem (3) reduces to a quadratic semi-definite programming problem with a large number of positive semi-definite matrices as the constrained variables. This is a challenging optimization problem without readily available solvers as most available semi-definite programming solvers such as SeDuMi [11] and SDPT3 [14] require the linear objective functions. Toh et al. [13] proposed an inexact primal-dual algorithm to solve a special class of convex quadratic semi-definite programming problem. However, their algorithm only deals with a single positive semi-definite matrix and cannot be directly applied to our optimization problem. Instead, we will exploit the special feature in our problem that $d = 3$ is a relatively small number and design a specific algorithm based on primal-dual path-following interior-point method to solve this subproblem.

The primal problem is given by

$$\begin{aligned} \min_{\mathbf{W}} \quad & \frac{1}{2} \sum_{i=1}^m \sum_{k=1}^n \|V_{ki} - \sum_{j=1}^r W_{kj} h_{ji}\|_F^2 \\ \text{s.t.} \quad & W_{kj} \succeq 0, k = 1, \dots, n, j = 1, \dots, r. \end{aligned} \quad (4)$$

We introduce the $d \times d$ symmetric matrices Z_{kj} associated with the matrix inequalities $W_{kj} \succeq 0$, where $k = 1, \dots, n$, $j = 1, \dots, r$. The Lagrangian of problem (4) is then

$$L(\mathbf{W}, \mathbf{Z}) = \frac{1}{2} \sum_{i=1}^m \sum_{k=1}^n \|V_{ki} - \sum_{j=1}^r W_{kj} h_{ji}\|_F^2 - \sum_{k=1}^n \sum_{j=1}^r \text{Tr}(Z_{kj} W_{kj}). \quad (5)$$

If we take the derivative of $L(\mathbf{W}, \mathbf{Z})$ with respect to W_{kj} and set it to zero, we have

$$Z_{kj} = - \sum_{i=1}^m (V_{ki} - \sum_{l=1}^r W_{kl} h_{li}) h_{ji}. \quad (6)$$

Substituting this expression back into the Lagrangian gives the dual problem

$$\begin{aligned} \max_{\mathbf{Z}} \quad & \sum_{i=1}^m \sum_{k=1}^n \left(-\frac{1}{2} \|V_{ki} - \sum_{j=1}^r W_{kj} h_{ji}\|_F^2 + \langle V_{ki} - \sum_{j=1}^r W_{kj} h_{ji}, V_{ki} \rangle \right) \\ \text{s.t.} \quad & \sum_{i=1}^m \left(V_{ki} h_{ji} - \sum_{l=1}^r W_{kl} h_{li} h_{ji} \right) + Z_{kj} = 0 \\ & Z_{kj} \succeq 0, k = 1, \dots, n, j = 1, \dots, r. \end{aligned} \quad (7)$$

For primal-dual interior point method, we use the perturbed Karush-Kuhn-Tucker (KKT) conditions:

$$\begin{aligned} Z_{kj} + \sum_{i=1}^m (V_{ki} - \sum_{l=1}^r W_{kl} h_{li}) h_{ji} &= 0 \\ W_{kj} Z_{kj} &= \nu_{kj} I \\ W_{kj} \succeq 0, Z_{kj} \succeq 0 \\ k &= 1, \dots, n, j = 1, \dots, r \end{aligned} \quad (8)$$

where ν_{kj} are positive parameters. Given the current iterate (\mathbf{W}, \mathbf{Z}) , the search direction $(\Delta \mathbf{W}, \Delta \mathbf{Z})$ at each interior-point iteration is the solution of the following Newton system

$$\sum_{l=1}^r \Delta W_{kl} \left(\sum_{i=1}^m h_{li} h_{ji} \right) - \Delta Z_{kj} = R_{d_{kj}} := Z_{kj} + \sum_{i=1}^m (V_{ki} - \sum_{l=1}^r W_{kl} h_{li}) h_{ji} \quad (9)$$

$$H_{P_{kj}} (\Delta W_{kj} Z_{kj} + W_{kj} \Delta Z_{kj}) = \sigma \mu_{kj} I - H_{P_{kj}} (W_{kj} Z_{kj} + \Delta W_{kj} \Delta Z_{kj}) \quad (10)$$

where $\mu_{kj} = \langle W_{kj}, Z_{kj} \rangle / d$ and $\sigma \in (0, 1)$ is the centering parameter. The symmetrization scheme H_P defined as $H_P(M) = \frac{1}{2}[PMP^{-1} + P^{-T}M^T P^T]$ is required here to generate symmetric ΔW_{kj} and ΔZ_{kj} . Inspired by [12], we choose $P_{kj} = T_{kj}^{-1/2}$, where T_{kj} is the Nesterov-Todd scaling matrix satisfying $T_{kj}Z_{kj}T_{kj} = W_{kj}$. The linearization of equation (10) gives

$$\mathcal{E}_{kj}(\Delta W_{kj}) + \mathcal{F}_{kj}(\Delta Z_{kj}) = R_{c_{kj}} := \sigma\mu_{kj}I - H_{P_{kj}}(W_{kj}Z_{kj}) \quad (11)$$

where \mathcal{E}_{kj} and \mathcal{F}_{kj} are linear operators. By eliminating ΔZ_{kj} in equations (9) and (11), we obtain

$$\sum_{l=1}^r \left(\sum_{i=1}^m h_{li} h_{ji} \right) \Delta W_{kl} + \mathcal{F}_{kj}^{-1} \mathcal{E}_{kj}(\Delta W_{kj}) = R_{d_{kj}} + \mathcal{F}_{kj}^{-1} R_{c_{kj}}. \quad (12)$$

Therefore, we just need to solve n linear systems to get $\Delta \mathbf{W}$. Each linear system includes $\frac{d(d+1)r}{2}$ equations. Because both d and r are small for our problem, these linear systems can be solved efficiently, and $\Delta \mathbf{Z}$ can be computed easily using equation (9). The detail steps of the algorithm is shown in Algorithm 2.

3.2 Optimization with respect to coefficient matrix \mathbf{H}

With a fixed \mathbf{W} , (3) becomes a convex optimization problem. Since

$$\frac{\partial E}{\partial h_{ji}} = - \sum_{k=1}^n \text{Tr}(V_{ki} W_{kj}) + \sum_{k=1}^n \sum_{l=1}^r \text{Tr}(W_{kj} W_{kl}) h_{li}, \quad (13)$$

setting $\frac{\partial E}{\partial h_{ji}} = 0$ gives

$$\sum_{l=1}^r \left(\sum_{k=1}^n \langle W_{kj}, W_{kl} \rangle \right) h_{li} = \sum_{k=1}^n \langle V_{ki}, W_{kj} \rangle. \quad (14)$$

Thus we just need to solve a non-negative least squares problem $\mathbf{A}_{r \times r} \mathbf{H}_{r \times m} = \mathbf{B}_{r \times m}$ to get h_{ji} , where $A_{jl} = \sum_{k=1}^n \langle W_{kj}, W_{kl} \rangle$ and $B_{ji} = \sum_{k=1}^n \langle V_{ki}, W_{kj} \rangle$, for all i, j, l . In our implementation, we use the fast active set method proposed by Van Benthem and Keenan [15] to solve this large-scale nonnegative least squares problem.

Once we have obtained the basis matrix \mathbf{W} , we can easily compute the ‘‘projection’’ of a new diffusion tensor image $\mathbf{X} = (X_1, \dots, X_n)^T$ by solving the following optimization problem

$$\min_{\mathbf{y} \geq 0} \sum_{k=1}^n \|X_k - \sum_{j=1}^r W_{kj} y_j\|_F^2 \quad (15)$$

where each X_k is a diffusion tensor and $\mathbf{y} = (y_1, \dots, y_r)^T$ is the nonnegative coefficient vector. This problem, similar to the subproblem with respect to \mathbf{H} above, can also be efficiently solved using the same fast active set method.

3.3 Diffusion tensor image segmentation

Because of its relations to K-means [2] and probabilistic latent semantic analysis [4], nonnegative matrix factorization has been widely used in data clustering, e.g., [19]. In this section, we formulate the diffusion tensor image segmentation problem as a special case of our more general nonnegative factorization problem with spatial constraints.

Algorithm 2 Quadratic semi-definite programming for nonnegative factorization

1: **Initialization** $W_{kj} \succeq 0, Z_{kj} \succeq 0, \forall k, j$ and $\tau = 0.9$.

2: **Convergence test** Stop the iteration if the accuracy measure ϕ is sufficiently small.

$$\phi = \frac{\sum_{k=1}^n \sum_{j=1}^r \langle W_{kj}, Z_{kj} \rangle}{1 + |\mathbf{pobj}| + |\mathbf{dobj}|}$$

where \mathbf{pobj} and \mathbf{dobj} are values of primal and dual objective functions.

3: **Predictor step** Compute the predictor search direction $(\delta\mathbf{W}, \delta\mathbf{Z})$ by choosing $\sigma = 0$.

4: **Predictor step-length** Compute $\alpha_p = \min(1, \tau\alpha)$. α is the maximum step length that can be taken so that for $k = 1, \dots, n$ and $j = 1, \dots, r$, $W_{kj} + \alpha\delta W_{kj}$ and $Z_{kj} + \alpha\delta Z_{kj}$ remain positive semidefinite.

5: **Centering rule** Set

$$\sigma = \frac{\sum_{k,j} \langle W_{kj} + \alpha_p \delta W_{kj}, Z_{kj} + \alpha_p \delta Z_{kj} \rangle}{\sum_{k,j} \langle W_{kj}, Z_{kj} \rangle}.$$

6: **Corrector step** Compute the search direction $(\Delta\mathbf{W}, \Delta\mathbf{Z})$ using

$$R_{c_{kj}} = \sigma \mu_{kj} I - H_{P_{kj}}(W_{kj} Z_{kj} + \delta W_{kj} \delta Z_{kj}).$$

7: **Corrector step-length** Compute α_c similar to step 4 with $(\delta\mathbf{W}, \delta\mathbf{Z})$ replaced by $(\Delta\mathbf{W}, \Delta\mathbf{Z})$.

8: Update (\mathbf{W}, \mathbf{Z}) to the next iterate $(\mathbf{W}^+, \mathbf{Z}^+)$.

$$W_{kj}^+ = W_{kj} + \alpha_c \Delta W_{kj}, \quad Z_{kj}^+ = Z_{kj} + \alpha_c \Delta Z_{kj}.$$

9: Update the step-length parameter by $\tau^+ = 0.9 + 0.08\alpha_c$.

Specifically, given a diffusion tensor image of size m , we arrange the m tensors in a row to form the data matrix $\mathbf{V} = (V_1, \dots, V_m)$. The original nonnegative factorization problem is modified as

$$\min_{\mathbf{W}, \mathbf{H}} \frac{1}{2} \sum_{i=1}^m \|V_i - \sum_{j=1}^r W_j h_{ji}\|_F^2 + \frac{\lambda}{2} \sum_{(k,l) \in \Omega} \sum_{j=1}^r (h_{jk} - h_{jl})^2 \quad (16)$$

with nonnegative constraints $W_j \succeq 0$ and $h_{ji} \geq 0$, for all i, j . The first term is simply the objective function in (3) given \mathbf{V} as a row vector. The second term is the spatial smoothness (soft) constraint that requires neighboring pixels to have similar coefficients, and in the equation above, Ω denotes the edge set of the discrete image graph, and λ the coupling parameter. The optimization problem (16) can be solved using the

same alternating method discussed above as the second term is also quadratic in \mathbf{H} . Once the coefficient matrix \mathbf{H} has been determined, we cluster the columns of \mathbf{H} to produce the diffusion tensor image segmentation. In our implementation, we use K-means for this last clustering step.

4 Experiments

In this section, we present two sets of experimental results. The first experiment is on diffusion tensor image segmentation using the segmentation algorithm outlined in the previous section. For the second set of experiments, we work with multiple images, and the results demonstrated that, as for scalar-valued images, meaningful parts can be discovered or detected using nonnegative factorization.

4.1 Diffusion tensor image segmentation

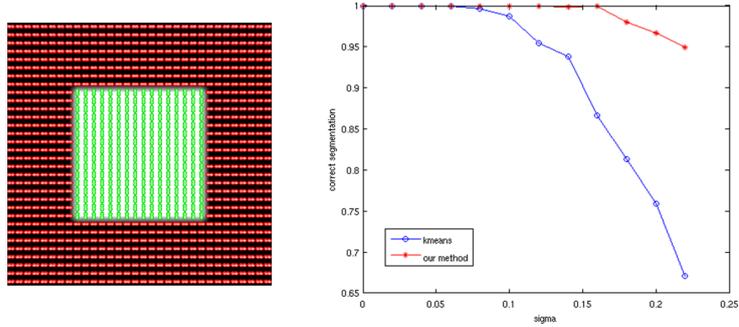


Fig. 1. **Left:** Input tensor field without noise. **Right:** Segmentation accuracy vs. different levels of added Gaussian noise $N(0, \sigma)$ with covariance σ .

Synthetic Tensor Images In this experiment, we test the accuracy of the segmentation algorithm using synthetic tensor images with various levels of added noise. We first generate the 32×32 tensor image \mathbf{T} shown in Figure 1 that contains only two tensors: diagonal matrices $diag(0.5, 0.25, 0.25)$ and $diag(0.25, 0.5, 0.25)$, and this defines the ground truth of the segmentation. Different levels of Gaussian noise $N(0, \sigma)$ are added to the tensor image \mathbf{T} to generate noisy tensor images, and we compare the segmentation accuracy of our method with the segmentation algorithm based on clustering the pixels using K-means (on the tensors). In this experiment, the segmentation accuracy is defined by the percentage of correctly segmented pixels, and the comparison across different noise levels is plotted in Figure 1. The result clearly shows the robustness of our method when compared with K-means, especially in the presence of substantial amount of noise.

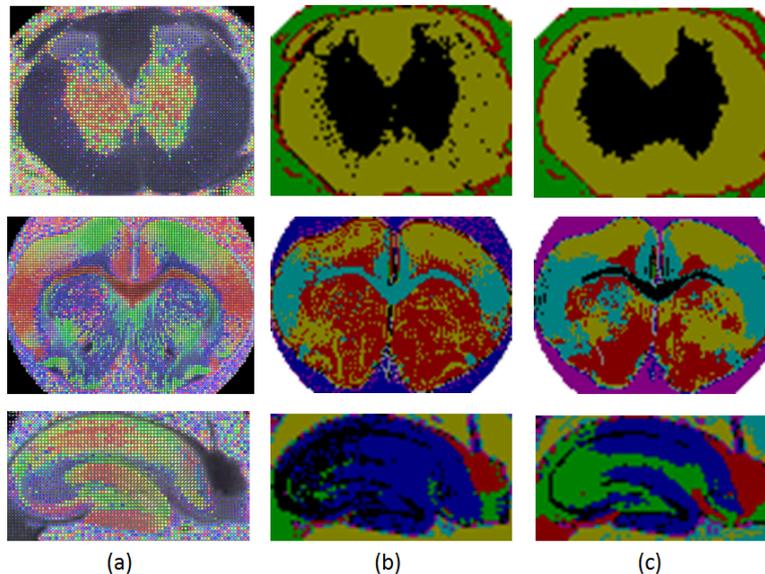


Fig. 2. **First Row:** Spinal cord of a rat. **Second Row:** Corpus callosum of a rat. **Third Row:** Hippocampus of a rat. Columns **(a):** Diffusion tensor images. **(b):** Segmentation results using K-means. **(c):** Segmentation results using our method. Segments are color-coded.

Diffusion Tensor Images We next present segmentation results on three real diffusion tensor images shown in Figure 2. These are DTI images of the spinal cord, corpus callosum and an isolated hippocampus of a rat. The data were acquired using a PGSE with $TR=1.5s$, $TE=28.3ms$, $bandwidth=35KHz$. Twenty-one diffusion weighted images with a b -value of $1250s/mm^2$ were collected. The sizes of the regions of interest for rat spinal cord, corpus callosum and hippocampus are 71×61 , 101×74 and 71×39 , respectively. In this experiment, the number of clusters for these images are four, eight and seven, respectively, and the number of basis elements (columns in \mathbf{W}) is set to five for all three images. Again, we compared our method with the K-means based segmentation and the results consistent demonstrate that our method can produce anatomically more accurate and meaningful segmentation than the straightforward K-means clustering. Finally, we note that both algorithms use K-means for clustering pixels. However, in our method, K-means is applied only to the coefficient vectors in \mathbf{H} , while in the comparison method, K-means is applied directly to the tensors. While the clustering power of the nonnegative factorization for matrices are well-known for scalar-valued data (e.g., [2]), our experimental results provide the first convincing evidence of its clustering power for tensor-valued data as the two sets of experiments have shown that there is no reasons to expect that a direct clustering on tensors would produce desired segmentation results. However, direct clustering on coefficient vectors does yield satisfactory results.

4.2 Nonnegative factorization with multiple tensor fields

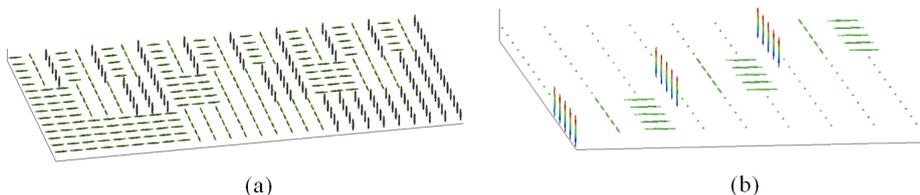


Fig. 3. (a) Visualization of the data matrix \mathbf{V} , each of whose columns represents a 5×3 tensor image. (b) Visualization of the basis matrix \mathbf{W} , each of whose columns represents a basis tensor image.

For a suitable collection of images, it is well-known that NMF has the ability to automatically determine similar decompositions of the images into their common parts and provide part-based representations for the images in the collection [3]. In this experiment using synthetic data, we show that nonnegative factorization also has similar capability for tensor images. Twenty-seven tensor fields with size 5×3 (15 pixels) are generated and form the columns of the data matrix \mathbf{V} as shown in Figure 3(a). We factor \mathbf{V} into the coefficient matrix \mathbf{H} and the basis matrix \mathbf{W} shown in Figure 3(b). Our factorization algorithm correctly determines the nine basis elements (columns of \mathbf{W}) required to form the data matrix \mathbf{V} , and the coefficient matrix returned by our algorithm is a sparse matrix. Furthermore, the L^2 factorization error is less than 8.57×10^{-10} .

Finally, we present a preliminary result on applying our nonnegative factorization method to automatically discover and segment anatomically important regions from a collection of 53 rat brain diffusion tensor images. In the preprocessing step, all images are aligned using similarity transforms, and in each image, a region of interest of size $46 \times 27 \times 7$ is manually selected. The left column of Figure 4 displays five sample slices from the input 3D diffusion tensor images. We apply the factorization algorithm with $r = 5$ (five basis elements) to this collection of DTIs, and one sample slice from each of the five basis images found by the algorithm is shown on the right column in Figure 4. Important anatomical regions such as white matter, putamen and nucleus accumbens are clearly represented in the five basis images.

5 Conclusions

This paper introduces the novel notion of nonnegative factorization of single and multiple tensor-valued images. The well-known method of nonnegative matrix factorization is extended to tensor-valued data, and an algorithm is proposed to efficiently and reliably solve the new factorization problem formulated as an optimization problem with a non-convex objective function. We have formulated a new approach to DTI segmentation using the proposed nonnegative factorization, and experimental results have shown

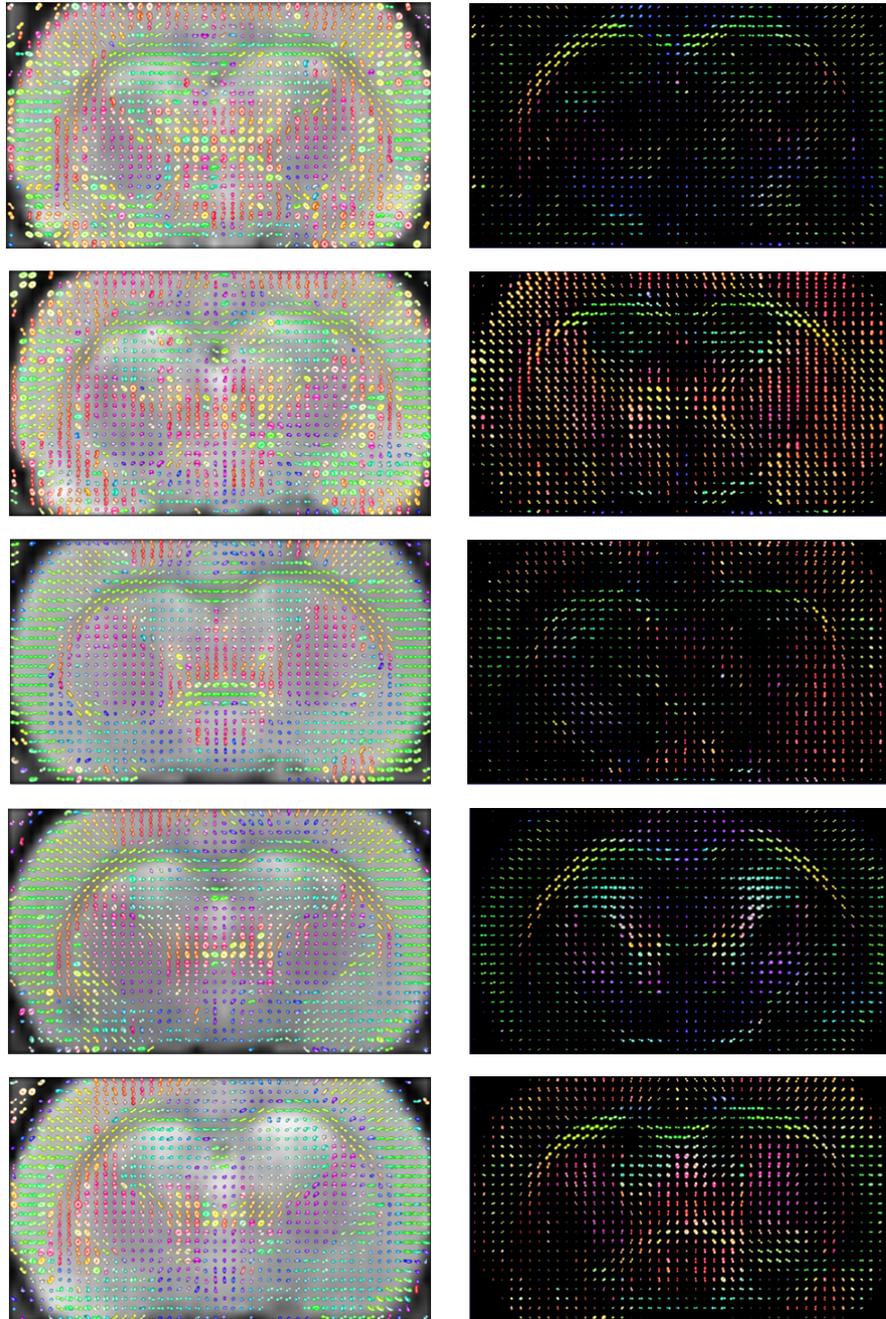


Fig. 4. Left column: Five sample slices from 53 input rat brain diffusion tensor images. **Right column:** Sample slices from the five basis tensor images produced by the proposed nonnegative factorization algorithm. We have enhanced the anisotropy for better visualization. The first, second and fourth basis images clearly include white matter, putamen and nucleus accumbens, respectively.

that our algorithm offers a competitive alternative to currently available methods in terms of its accuracy and efficiency. Perhaps more importantly, our work has demonstrated the usefulness and versatility of the notion of nonnegative factorization, now in the more general setting of tensor-valued images. We believe that this simple yet powerful notion will find its rightful place in the analysis of tensor-valued images. For example, it could potentially offer a new and effective approach to simultaneously analyze a large number of diffusion tensor images, an essential task in group studies and other applications, that can currently be achieved using only a limited number of available tools, such as the standard statistical analysis in Euclidean space or principal geodesic analysis on Riemannian manifolds [18].

References

1. Boyd, S., Vandenberghe, L.: Convex optimization. Cambridge University Press (2004)
2. Ding, C., He, X., Simon, H.: On the equivalence of nonnegative matrix factorization and spectral clustering. In: Proc. SIAM Data Mining Conf (2005)
3. Donoho, D., Stodden, V.: When does non-negative matrix factorization give a correct decomposition into parts? In: NIPS (2004)
4. Gaussier, E., Goutte, C.: Relation between PLSA and NMF and implications. In: ACM SIGIR (2005)
5. Grippo, L., Sciandrone, M.: On the convergence of the block nonlinear gauss-seidel method under convex constraints. Operations Research Letters 26(3), 127–136 (2000)
6. Joshi, S., Karthikeyan, S., Manjunath, B., Grafton, S., Kiehl, K.: Anatomical parts-based regression using non-negative matrix factorization. In: CVPR (2010)
7. Lee, D., Seung, H.: Learning the parts of objects by non-negative matrix factorization. Nature 401, 788–791 (1999)
8. Lenglet, C., Rousson, M., Deriche, R.: DTI segmentation by statistical surface evolution. IEEE Trans. on Medical Imaging 25(6), 685–700 (2006)
9. Li, S., Hou, X., Zhang, H., Cheng, Q.: Learning spatially localized, parts-based representation. In: CVPR (2001)
10. Nocedal, J., Wright, S.: Numerical optimization. Springer (2000)
11. Sturm, J.: Using SeDuMi 1.02, a MATLAB toolbox for optimization over symmetric cones. Optimization methods and software 11(1), 625–653 (1999)
12. Todd, M., Toh, K., Tutuncu, R.: On the Nesterov-Todd direction in semidefinite programming. SIAM Journal on Optimization 8(3), 769–796 (1998)
13. Toh, K., Tutuncu, R., Todd, M.: Inexact primal-dual path-following algorithms for a special class of convex quadratic SDP and related problems. Pac. J. Optim 3, 135–164 (2007)
14. Tutuncu, R., Toh, K., Todd, M.: Solving semidefinite-quadratic-linear programs using SDPT3. Mathematical Programming 95(2), 189–217 (2003)
15. Van Benthem, M., Keenan, M.: Fast algorithm for the solution of large-scale non-negativity-constrained least squares problems. Journal of chemometrics 18(10), 441–450 (2004)
16. Wang, Z., Vemuri, B.: DTI segmentation using an information theoretic tensor dissimilarity measure. IEEE Trans. on Medical Imaging 24(10), 1267–1277 (2005)
17. Weldelesassie, Y., Hamarneh, G.: DT-MRI segmentation using graph cuts. In: Proc. of SPIE Medical Imaging: Image Processing (2007)
18. Xie, Y., Vemuri, B., Ho, J.: Statistical analysis of tensor fields. In: MICCAI (2010)
19. Xu, W., Liu, X., Gong, Y.: Document clustering based on non-negative matrix factorization. In: ACM SIGIR (2003)