Numerical Optimization 2018.
Tensor Fitting in Diffusion Tensor Imaging

François Lauze, Oswin Krause

March 5, 2018

1 Introduction

Diffusion Weighted Imaging (DWI)is an in-vivo, non invasive Magnetic Resonance Imaging (MRI) modality used to characterize orientation properties of different tissues, and is especially focused towards muscle fiber and brain white matter structure.

In DWI, some information about how water diffuses at a given location is obtained by probing magnetic properties of water molecules in a series of directions. When applying an oriented magnetization in a given direction, it roughly measures the loss of magnetization after a short period of time, and relates this loss to the potential diffusion of water in that direction. The more water diffuses, the larger the loss will be. This is formalized by the *Stejskal-Tanner equation* (1965) in the simplified form proposed by D. Le Bihan (1985). It reads

$$s_i^p = s_0^p e^{-bA_i} \tag{1.1}$$

where

- p is the 3D location probed,
- i is the direction probed, corresponding to a point u_i on the unit sphere $\mathbb{S}^2 \subset \mathbb{R}^3$, i.e. a unit vector in $\mathbb{R}^3$
- s_i^p is the magnetization signal read at this location and direction,
- s_p^0 is a *baseline* magnetization signal at p,
- b is a (complicated) positive parameter encoding a lot of information about the intensity, shape and duration of magnetization applied to obtained the measurement and A_i is the *apparent diffusion coefficient* along the direction u_i.

It is assumed that water diffuses more in the main direction of an anatomical structure, such as an axon or a muscle fiber, than across it, as water motion would be hindered by cellular membranes or the myeline sheath around an axon, though it is also expected that this cross diffusivity is never fully hindered. In

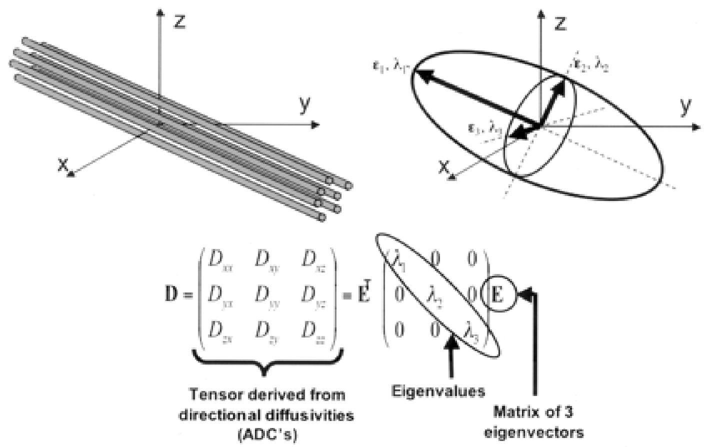

Figure 1: Diffusion along fibers (from Jellison, Field, Medow, Lazar, Salamat and Alexander, American Journal of Neuroradiology March 2004, 25 (3) 356-369)

Diffusion Tensor Imaging, the *apparent diffusion coefficient* A_i is assumed to have the form $u_i^\top D u_i$ where D is a 3×3 positive definite matrix, also called *tensor* (P. Basser, 1994).

$$D = \begin{pmatrix} D_{11} & D_{12} & D_{13} \\ D_{12} & D_{22} & D_{23} \\ D_{13} & D_{23} & D_{33} \end{pmatrix}$$

This is linked to an assumption of Gaussianity of the diffusion process. Note that because D is symmetric, it is determined by 6, and not 9 coefficients.

During an imaging session, series of volumetric images are acquired and one volumetric image represents measurements at all voxels for a given direction, or a "baseline" measurement used to estimate the volume of parameters s_0^p.

2 Tensor Fitting

We formulate here the problem of tensor fitting as a non linear least square regression problem. Given n directions $u_1, \ldots, u_n$, n measured values $s_1, \ldots, s_n$, the baseline value s_0 and b, find the diffusion tensor D which minimizes

$$F(D) = \frac{1}{2} \sum_{j=1}^{n} \left(s_j - s_0 e^{-b u_j^T D u_j} \right)^2 = \frac{1}{2} \sum_{j=1}^{n} r_j(D)^2. \tag{2.1}$$

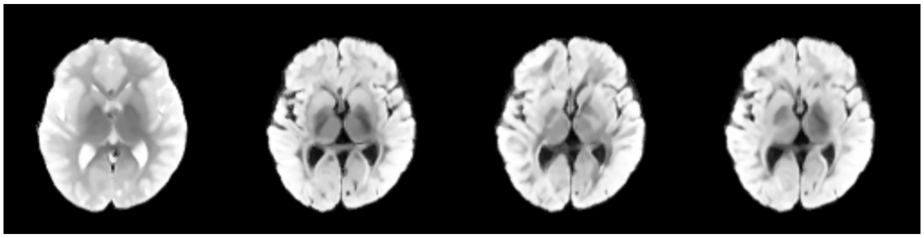

Figure 2: DWI measurements at 4 different directions, from the ADNI Database

Of course the values s_j, $j = 0 \dots n$, are given for a particular voxel. Formally, we should add the constraint that all the eigenvalues of D are positive,

$$\lambda_1(D) \geq \lambda_2(D) \geq \lambda_3(D) > 0$$

but we will ignore them here[1].

In the assignment, you will have to implement tensor fitting using a Levenberg-Marquardt approach. As help, we provide the gradient of F.

Gradient of F. The gradient is given by

$$\nabla F(D) = s_0 b \sum_{j=1}^{n} \left[\left(s_j - s_0 e^{-b u_j^T D u_j} \right) e^{-b u_j^T D u_j} \underbrace{u_j u_j^\top}_{3\times 3 \text{ symmetric matrix}} \right]. \tag{2.2}$$

This can also be seen as a (matrix/vector)-valued function of the 6 parameters which describe D $D_{11}, D_{12}, D_{13}, D_{22}, D_{23}, D_{33}$i.e., a function on (an open subset of) $\mathbb{R}^6$. For that purpose, we introduce $d_1 = D_{11}, d_2 = D_{12}, d_3 = D_{13}, d_4 = D_{22}, d_5 = D_{23}$ and $d_6 = D_{33}$ so that we can rewrite F as

$$F(D) = F(d_1, \dots . d_6) = \frac{1}{2} \sum_{j=1}^{n} \left(s_j - s_0 e^{-bL(u_j, d_1, \dots, d_6)} \right)^2$$

with $u_j = (u_{j1}, u_{j2}, u_{j3})^\top$ and $L(u_j, d_1, \dots, d_6)$ is the quadratic form

$$L(u_j, d_1, \dots, d_6) = d_1 u_{j1}^2 + 2d_2 u_{j1} u_{j2} + 2d_3 u_{j1} u_{j3} + d_4 u_{j2}^2 + 2d_5 u_{j2} u_{j3} + d_6 u_{j3}^2.$$

As D is 6th-dimensional, the gradient $\nabla F(D)$ can be written as a vector of $\mathbb{R}^6$. We do it below, though the Levenberg-Marquardt algorithm can be implemented with the more intrinsic form (2.2). This will help us to identify the matrix $J(D)$.

[1] A way to enforce recovery of a SPD matrix is to write $D = e^E$ where E is symmetric and e^E is the matrix exponential $\sum_{k=0}^{\infty} \frac{E^k}{k!}$. D is guaranteed to be SPD and one may want to optimize $G(E) = F(e^E) = \frac{1}{2} \sum_{j=1}^{n} \left(s_j - s_0 e^{-b u_j^T e^E u_j} \right)^2$ instead of $F(D)$. This represents an extra layer of calculations but can be solved with Levenberg-Marquardt-like approaches too.

The vectorial part in (2.2) is of the form $uu^\top$ with $u = (u_1, u_2, u_3)$. A direct calculation gives

$$uu^\top = \begin{pmatrix} {u_1}^2 & u_1u_2 & u_1u_3 \\ u_1u_2 & {u_2}^2 & u_2u_3 \\ u_1u_2 & u_2u_3 & {u_3}^2 \end{pmatrix} \rightsquigarrow (u_1^2, u_1u_2, u_1u_3, u_2^2, u_2u_3, u_3^2)^\top \in \mathbb{R}^6$$

i.e., as a symmetric (positive semi-definite) matrix, $u^\top u$ is represented by the right hand side for the 6 parameters representation discussed above. From that, a parametric form (as opposed to matricial) for the gradient is given by

$$\nabla F(d_1, \ldots, d_6) = s_0 b \sum_{j=1}^{n} e^{-bu_j^T Du_j} \begin{pmatrix} u_{i1}^2 \\ u_{i1}u_{i2} \\ u_{i1}u_{i3} \\ u_{i2}^2 \\ u_{i2}u_{i3} \\ u_{i_3}^2 \end{pmatrix} r_j(D) \tag{2.3}$$

from which we get that

$$\begin{aligned} J(D) &= J(d_1 \ldots, d_6) \\ &= s_0 b \begin{pmatrix} \ell_1 u_{11}^2 & \ell_1 u_{11}u_{12} & \ell_1 u_{11}u_{13} & \ell_1 u_{12}^2 & \ell_1 u_{12}u_{13} & \ell_1 u_{13}^2 \\ \ell_2 u_{21}^2 & \ell_2 u_{21}u_{22} & \ell_2 u_{21}u_{23} & \ell_2 u_{22}^2 & \ell_2 u_{22}u_{23} & \ell_2 u_{23}^2 \\ \vdots & & & & & \vdots \\ \ell_2 u_{n1}^2 & \ell_n u_{n1}u_{n2} & \ell_n u_{n1}u_{n3} & \ell_n u_{n2}^2 & \ell_n u_{n2}u_{n3} & \ell_n u_{n3}^2 \end{pmatrix} \end{aligned}$$

with $\ell_j = e^{-bu_j^\top Du_j}$.

Note that in that representation $J(D) \in \mathbb{R}^{n\times 6}$.

Hessian of F. The Hessian is a quadratic form, which algebraically reads

$$\operatorname{Hess} F(D)(H, H) = s_0 b^2 \sum_{j=1}^{n} \left(2s_0 e^{-bu_j^\top Du_j} - s_j\right) e^{-bu_j^\top Du_j} \left(u_j^\top H u_j\right)^2 \tag{2.4}$$

H is a 3×3 symmetric matrix and by parametrizing it as

$$H = \begin{pmatrix} h_1 & h_2 & h_3 \\ h_2 & h_4 & h_5 \\ h_3 & h_5 & h_6 \end{pmatrix}$$

the expression above can be rewritten as

$$\begin{aligned} \operatorname{Hess} F(D)(\mathbf{h}, \mathbf{h}) &= \mathbf{h}^\top B(D)\mathbf{h}, \quad \mathbf{h} = (h_1, h_2, h_3, h_4, h_5, h_6)^\top. \\ &= \sum_{i,j=1}^{6} B(D)_{ij} h_i h_j \end{aligned} \tag{2.5}$$

and $B(D)$ is the Hessian matrix for the parametrization $(h_1, h_2, h_3, h_4, h_5, h_6)$. Identifying the components of $B(D)$ from (2.4) is tedious, though straightforward. In order to perform this rewriting, (2.4) contains expressions of the form

$$\left(u^\top H u\right)^2, \quad u = (u_1, u_2, u_3)^\top$$

and after carefull expansion, it can be written as the quadratic form $\mathbf{h}^\top \mathbf{U} \mathbf{h}$, ($\mathbf{h} = (h_1, \dots, h_6)^\top$), with

$$\mathbf{U} = \begin{pmatrix} u_1^4 & 2u_1^3u_2 & 2u_1^3u_3 & u_1^2u_2^2 & 2u_1^2u_2u_3 & u_1^2u_3^2 \\ 2u_1^3u_2 & 4u_1^2u_2^2 & 4u_1^2u_2u_3 & 2u_1u_2^3 & 4u_1u_2^2u_3 & 2u_1u_2u_3^2 \\ 2u_1^3u_3 & 4u_1^2u_2u_3 & 4u_1^2u_3^2 & 2u_1u_2^2u_3 & 4u_1u_2u_3^2 & 2u_1u_3^3 \\ u_1^2u_2^2 & 2u_1u_2^3 & 2u_1u_2^2u_3 & u_2^4 & 2u_2^3u_3 & u_2^2u_3^2 \\ 2u_1^2u_2u_3 & 4u_1u_2^2u_3 & 4u_1u_2u_3^2 & 2u_2^3u_3 & 4u_2^2u_3^2 & 2u_2u_3^3 \\ u_1^2u_3^2 & 2u_1u_2u_3^2 & 2u_1u_3^3 & u_2^2u_3^2 & 2u_2u_3^3 & u_3^4 \end{pmatrix} \tag{2.6}$$

The Hessian matrix $B(D)$ (2.5) is then given by

$$B(D) = s_0 b^2 \sum_{j=1}^{n} \left(2s_0 e^{-bu_j^\top D u_j} - s_j\right) e^{-bu_j^\top D u_j} \mathbf{U}_j$$

where $\mathbf{U}_j$ is the matrix $\mathbf{U}$ with (u_1, u_2, u_3) substitued by (u_{j1}, u_{j2}, u_{j3}). Available from Absalon is a Matlab function `fhessian_component.m`, which computes the matrix $\mathbf{U}$ from a given vector $u = (u_1, u_2, u_3)^\top$). With it, getting the Hessian of F inn its 6th-dimensional parametrization should be easy.

Theoretically, 6 measurements should be enough to compute a diffusion tensor, and in the early period of DWI, this was indeed the case, only 6 or 7 measurements where acquired, due to time constraint and patient comfort. For some routine clinical use of DWI, this may still be the case at many Magnetic Resonance sites. Nowadays, some clinical and research protocols can allow many more directions. Clearly an increase in the number of measurements should also increase the fitting quality. One potential test is therefore to run the fitting algorithm with variable n and observe how it impacts the quality of the estimation of tensor D.

In order to represent a solution, which is a 3×3 SPD matrix D, and thus a 6-dimensional object, one often represent tensors as ellipsoids, as in 1. The main axes are proportional to the square-roots of the eigenvalues of D, and are oriented in the eigenvector directions of D. A Matlab function `drawtensor.m`, which plots a tensor as an ellipsoid is included in Absalon. Some explanation is provided in Sec 4.

3 Data

The data consists of a Matlab mat-file `dwidat.mat` with the following content

1. a variable called `bvecs`, which is 90x3 array each line encode a unit vector $u_i \in \mathbb{S}^2$ used to acquire directional information,

2. A variable `meas`, which is a 50x91 array which contains data from 50 voxels extracted from a scan of the Human Connectom Project database, a freely available database of MR data developed for research on human brain connectivity. Each line contains the values $(s_0, s_1, \dots, s_{90})$.

3. a double precision variable `b`, with value 1000.

4 Tensors and Ellipsoids

A standard way to plot a diffusion tensor D is to plot the implicit surface given by the equation

$$\mathcal{S} = \{\mathbf{x} \in \mathbb{R}^3, \quad \mathbf{x}^\top D^{-1}\mathbf{x} = 1\}. \tag{4.1}$$

An ellipsoid centered at 0, with major axes coinciding with the standard axes has the Cartesian equation and half extent in x: $\sqrt{\lambda_1}$, half extent in y: $\sqrt{\lambda_2}$, half extent in z: $\sqrt{\lambda_3}$, i.e. containing the points $\pm\sqrt{\lambda_i}e_i$, is given by

$$\frac{x^2}{\lambda_1} + \frac{y^2}{\lambda_2} + \frac{z^2}{\lambda_3} = 1$$

This can be rewritten as

$$\mathbf{x}^\top \Lambda^{-1}\mathbf{x} = 1, \quad \mathbf{x} = (x, y, z)^\top, \quad \Lambda = \begin{pmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{pmatrix}$$

Given a SPD matrix D, i.e., a tensor for us, it admits an eigendecomposition

$$D = R\Lambda R^\top$$

with R a rotation matrix and λ a diagonal matrix. Clearly $D^{-1} = R\Lambda^{-1}R^\top$. Take $\mathbf{y}$ which satisfies $\mathbf{y}^\top D^{-1}\mathbf{y} = 1$, and set $\mathbf{x} = R^\top\mathbf{y} = R^{-1}\mathbf{y} \equiv \mathbf{y} = R\mathbf{x}$. Then

$$\begin{aligned} 1 &= \mathbf{y}^\top D^{-1}\mathbf{y} = \mathbf{x}R^\top R\lambda^{-1}R^\top R\mathbf{x} \\ &= \mathbf{x}^\top \Lambda^{-1}\mathbf{x} \end{aligned}$$

and $\mathbf{x}$ is on the ellipsoid with eccentricities the square-roots of the eigenvalues of D and aligned with the standard axes. The set of $\mathbf{y} \in \mathbb{R}^3$ such that $\mathbf{y}^\top D^{-1}\mathbf{y} = 1$ is thus also an ellipsoid with eccentricities the square-roots of the eigenvalues of D, bu with main axes directions provided by R.

For instance, consider the basis

$$v_1 = \frac{1}{\sqrt{3}}\begin{pmatrix} 1 \\ 1 \\ 1 \end{pmatrix}, v_2 = \frac{1}{\sqrt{2}}\begin{pmatrix} 0 \\ 1 \\ -1 \end{pmatrix}, v_3 = \frac{1}{\sqrt{6}}\begin{pmatrix} -1 \\ 2 \\ -1 \end{pmatrix}.$$

It is orthonormal and thus the matrix $S = (v_1, v_2, v_3)$ is a rotation and $R = S^{-1} = S^T$ will rotate vectors v_1 to standard basis vector e_1, v_2 to standard basis vector e_2 and v_3 to standard basis vector e_3. Then set

$$B = R \begin{pmatrix} 36 & 0 & 0 \\ 0 & 4 & 0 \\ 0 & 0 & 1 \end{pmatrix} R^\top.$$

We thus expect the set $\mathbf{x}^\top B\mathbf{x} = 1$to be an ellipsoid with axes aligned resp. with v_1, v_2 and v_3 and eccentricities $(6, 2, 1)$ resp.

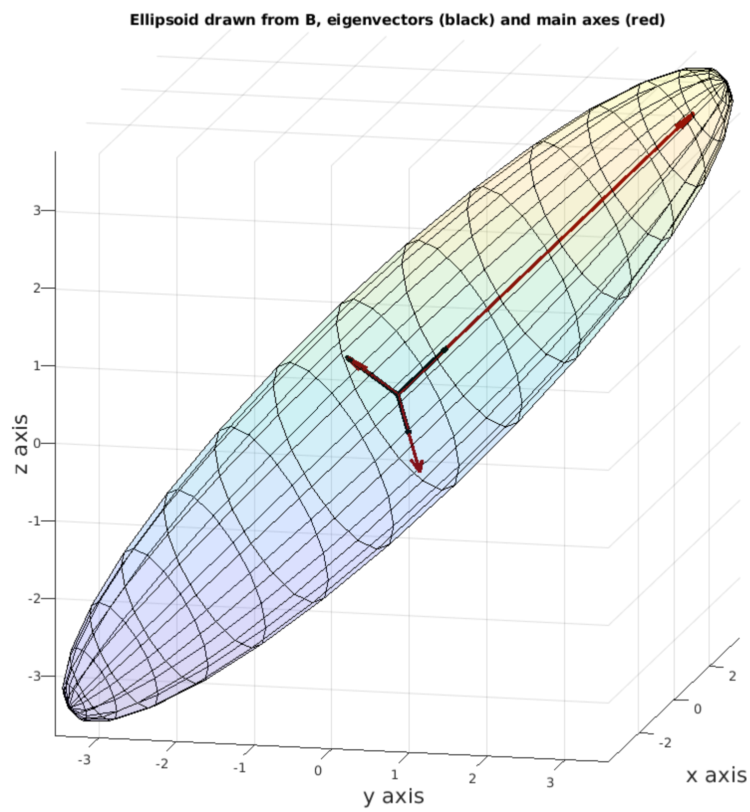

Figure 3: Ellipsoid $\mathbf{x}^\top B\mathbf{x} = 1$ drawn with the `drawtensor.m` function.