

Confidence and Uncertainty

Daniel Alexander

Recap

- Parametric mapping and DTI
- Linear least-squares estimate
- Maximum likelihood estimation
- Weighted linear least squares
- Non-linear fit
- Optimization
- Global and local maxima
- Constraints

Weighted linear least squares

- Errors on each measurement are independent but have different variance

$$\tilde{\mathbf{x}} = \underset{\mathbf{x}}{\operatorname{argmax}} \sum_{i=1}^K \frac{(A(\mathbf{y}_i) - S(\mathbf{x}; \mathbf{y}_i))^2}{\sigma_i^2}$$

- The maximum likelihood estimate is

$$\tilde{\mathbf{x}} = (G^T W G)^{-1} G^T W \mathbf{A}$$

$$\text{where } W_{ii} = \frac{1}{\sigma_i^2}$$

Error propagation

- Measurement A has standard deviation σ .
- What is the standard deviation of $f(A)$?

$$\sigma_f = \sigma \frac{\partial f}{\partial A}$$

- Simple version of more general approach called *error propagation*.

Non-linear DT fit (naïve!!)

```
voxdw = dwis(:,66,58,25);
ps = pinv(X)*log(voxdw);
startx = ps;
startx(1) = exp(startx(1));
h = optimset();
[par,RES,EXIT,OUT]=fminunc('DT_SSD',startx,h,voxdw,bvals,qhat);
```

```
function [sumRes, resJ] = DT_SSD(x, meas, bvals, qhat)
```

```
S0 = x(1);
Dxx=x(2);Dxy=x(3);Dxz=x(4);Dyy=x(5);Dyz=x(6);Dzz=x(7);
D = [[Dxx Dxy Dxz];[Dxy Dyy Dyz];[Dxz Dyz Dzz]]
```

```
% The abs is a cheat to add numerical stability.
S = S0*exp(-abs(sum(qhat.*(D*qhat)).*bvals))
```

```
sumRes = sum((meas - S').^2);
```

5

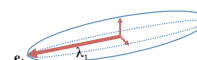
The Diffusion Tensor

- Proportional to the covariance of a trivariate Gaussian.

- Positive definite, symmetric 3x3 matrix:

$$D = \begin{bmatrix} D_{xx} & D_{xy} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{bmatrix} = \begin{bmatrix} \mathbf{e}_1 & \mathbf{e}_2 & \mathbf{e}_3 \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 & 0 \\ 0 & \lambda_2 & 0 \\ 0 & 0 & \lambda_3 \end{bmatrix} \begin{bmatrix} \mathbf{e}_1 \\ \mathbf{e}_2 \\ \mathbf{e}_3 \end{bmatrix}$$

- Determines the elliptical shape and orientation of the contours



Bayesian inference

- The maximum likelihood estimate maximizes $p(\mathbf{A} \mid \mathbf{x})$.
- Bayesian approach maximizes

Likelihood

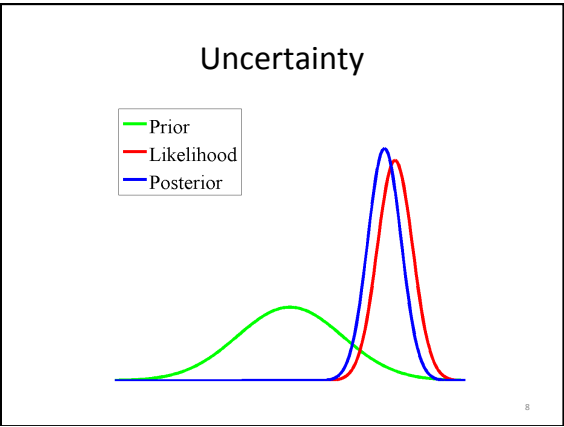
Posterior

Prior

$$p(\mathbf{x} \mid \mathbf{A}) = \frac{p(\mathbf{A} \mid \mathbf{x})p(\mathbf{x})}{p(\mathbf{A})} = \frac{p(\mathbf{A} \mid \mathbf{x})p(\mathbf{x})}{\int p(\mathbf{A} \mid \mathbf{x})p(\mathbf{x})d\mathbf{x}}$$

- The maximum a-posteriori (MAP) estimate is $\tilde{\mathbf{x}} = \operatorname{argmax} p(\mathbf{x} \mid \mathbf{A}) = \operatorname{argmax} (\log p(\mathbf{x} \mid \mathbf{A}) + \log p(\mathbf{x}))$

7



Uncertainty

- We would like to characterize the confidence or uncertainty in parameter estimates.
- Provide confidence intervals on \mathbf{x} .
- Sample the distribution on \mathbf{x} .

- Why?
- Laplace’s method
- Bootstrap resampling techniques
 - Parametric
 - Non-parametric
 - Residual/Wild
- Markov Chain Monte Carlo (MCMC)

9

Basic parametric mapping

ELSEVIER

NeuroImage

www.elsevier.com/locate/yimg
NeuroImage 27 (2005) 436 – 441

Three-dimensional quantitative magnetisation transfer imaging of the human brain

Mara Cercignani,^{a,*} Mark R. Symms,^b Klaus Schmierer,^a Philip A. Boulby,^b Daniel J. Tozer,^a Maria Ron,^a Paul S. Tofts,^a and Gareth J. Barker^c

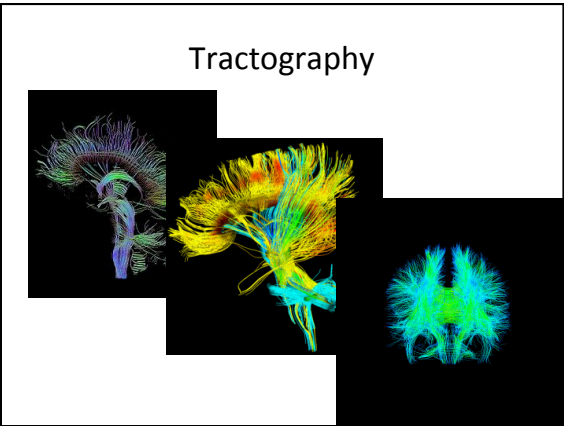
10

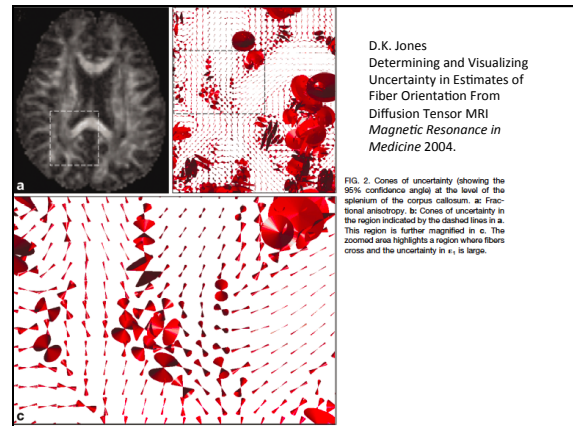
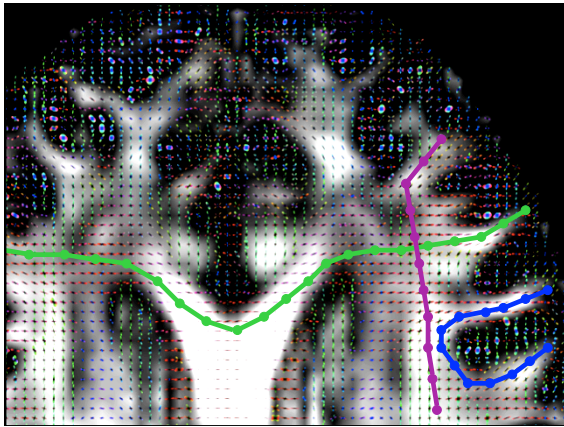
Eg: quantitative magnetization transfer study in the normal human brain.

Table 1
Average quantitative MT parameters from the 3D-MTSPGR sequence, measured in the brain of six healthy volunteers

Area		f [%]	T_2^* [ms]	T_2^* [ms]	T_{map} [ms]	T_2^* [ms]
Corpus callosum	Genu	Mean	10.1	11.4	726	728
		SD	0.8	0.8	68	68
	Splenium	Mean	9.7	11.9	711	713
		SD	1.2	0.4	83	83
Body	Genu	Mean	8.0	11.4	853	854
		SD	0.8	1.0	74	74
	Splenium	Mean	9.0	12.7	749	751
		SD	1.2	1.2	67	67
Internal capsule	Genu	Mean	9.3	13.8	734	736
		SD	1.1	8.9	47	47
	Splenium	Mean	8.8	13.7	730	732
		SD	0.7	7.9	46	46
Corona radiata	Genu	Mean	9.4	12.3	661	663
		SD	0.9	8.4	44	44
	Splenium	Mean	9.2	13.4	745	747
		SD	1.3	0.9	61	60
Caudate	Genu	Mean	9.4	11.8	718	719
		SD	0.8	0.5	51	51
	Splenium	Mean	8.1	11.7	825	827
		SD	0.9	1.1	70	70
Putamen	Genu	Mean	9.1	13.4	719	720
		SD	0.6	0.8	41	40
	Splenium	Mean	6.6	11.8	975	976
		SD	0.7	0.8	61	61
Frontal GM	Genu	Mean	5.0	11.4	1164	1164
		SD	0.6	0.8	117	117
	Splenium	Mean	5.2	10.9	1092	1092
		SD	0.7	0.6	85	85
Frontal WM	Genu	Mean	4.7	11.8	1279	1279
		SD	0.6	1.6	128	128
	Splenium	Mean	4.7	11.8	1279	1279
		SD	0.6	1.6	128	128

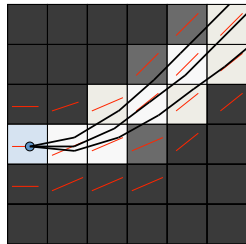
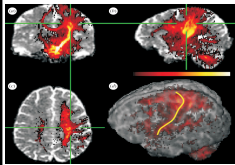
f = macromolecular proton fraction; A = free pool; B = restricted pool; SD = standard deviation; GM = grey matter; WM = white matter. See text for further details.
With the exception of the corpus callosum and the pons, the values were measured bilaterally.





Probabilistic Tractography

- Model uncertainty of fiber orientations
- Repeat:
 - Sample orientations
 - Track a streamline
- Count streamlines in each voxel



This is the basic PICO algorithm by Parker et al, JMIR 2003.

Laplace's method

- Generally, a method to approximate integrals.
- In our context: approximates the parameter distribution as a Gaussian.
- The mean is the ML and MAP estimate.
- The covariance is

$$\Sigma = \left(\frac{d^2}{d\mathbf{x}^2} \log p(\tilde{\mathbf{x}} | \mathbf{A}) \right)^{-1} = (H(\tilde{\mathbf{x}}))^{-1}$$
- H comes directly from fminunc
- Diagonal elements provide 2σ range.

16

Bootstrapping

- Use the data to guess its own distribution.
- Use the data distribution to estimate the parameter distribution.

Reminder of terms

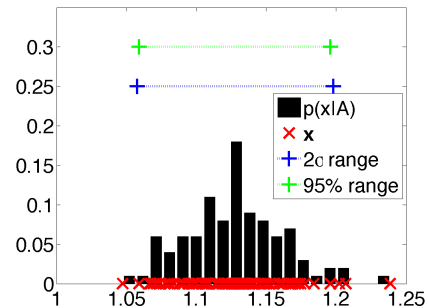
$$\mathbf{A} = \begin{pmatrix} A_1 \\ A_2 \\ \vdots \\ A_K \end{pmatrix} \quad \text{Data} \qquad \mathbf{S} = \begin{pmatrix} S(\mathbf{y}_1; \tilde{\mathbf{x}}) \\ S(\mathbf{y}_2; \tilde{\mathbf{x}}) \\ \vdots \\ S(\mathbf{y}_K; \tilde{\mathbf{x}}) \end{pmatrix} \quad \text{Model signals}$$

$$\tilde{\mathbf{x}} = \begin{pmatrix} x_1 \\ x_2 \\ \vdots \\ x_N \end{pmatrix} \quad \text{Parameter estimate} \qquad \mathbf{r} = \begin{pmatrix} A_1 - S(\mathbf{y}_1; \tilde{\mathbf{x}}) \\ A_2 - S(\mathbf{y}_2; \tilde{\mathbf{x}}) \\ \vdots \\ A_K - S(\mathbf{y}_K; \tilde{\mathbf{x}}) \end{pmatrix} \quad \text{Residuals}$$

Parametric bootstrap

- Fit model parameters
- Compute model signals
- Estimate noise parameters, eg $\tilde{\sigma}^2 = \frac{1}{K-1} \sum_{i=1}^K r_i^2$
- for $i=1:T$
 - Sample $\mathbf{E} = v_1, \dots, v_K$ from noise distribution eg $N(0, \tilde{\sigma})$
 - Synthesize bootstrap data set $\hat{\mathbf{A}}_i = \mathbf{S} + \mathbf{E}$
 - Estimate best fit \mathbf{x}_i to $\hat{\mathbf{A}}_i$
- Output samples $\mathbf{x}_1, \dots, \mathbf{x}_T$ of $p(\mathbf{x} | \mathbf{A})$.

Example output



Bootstrap samples

- Compute confidence intervals
 - 2σ range appropriate if the distribution is Gaussian
 - 95% range is from the 0.025 T -th sample to the 0.975 T -th sample.
- Allow us to visualize the distribution on \mathbf{x} .

Non-parametric bootstrap

- Resample data directly
- In each variant, the overall algorithm is as for parametric bootstrap, but the method for generating the bootstrap sample differs

Classical bootstrap

- Samples from the original measurements with replacement:

$$\hat{A}_{ij} = A_{[U(1,K+1)]}$$

- The subscript on the right hand side is a uniformly distributed integer in the range $[1, K+1]$.

Repetition bootstrap

- Requires several, R , repeat measurements:

$$\begin{pmatrix} A_{11} & A_{12} & \dots & A_{1R} \\ A_{21} & A_{22} & \dots & A_{2R} \\ \vdots & \vdots & \ddots & \vdots \\ A_{K1} & A_{K2} & \dots & A_{KR} \end{pmatrix}$$

- Each bootstrap data set picks one from each $(A_{k1}, A_{k2}, \dots, A_{kR})$.

$$\hat{A}_{ij} = A_{[U(1,R+1)]}$$

Residual bootstrap

- Resamples the residuals rather than the data
- For each bootstrap data point select one of the residuals r_1, \dots, r_K at random and add it to the corresponding model signal.

$$\hat{A}_{ij} = S(y_j; \tilde{\mathbf{x}}) + r_{[U(1,K+1)]}$$

Wild bootstrap

- Multiplies each residual by a standard normal distribution

$$\hat{A}_{ij} = S(y_j; \tilde{\mathbf{x}}) + r_j N(0,1)$$

Issues with bootstrap

- Which kind of bootstrap?
 - Parametric bootstrap assumes model is correct
 - Classical bootstrap disrupts experiment design
 - Repetition bootstrap is expensive
 - Residual and wild bootstrap retain experiment design economically, but assume signal model.
 - Wild bootstrap accommodates different noise models on different data points
- How many bootstrap samples do I need?
 - Monitor convergence of statistic of interest

MCMC

\mathbf{x}_0 = start point

$p_0 = p(\mathbf{x}_0 | \mathbf{A})$

for $i=1 \dots T$

$\mathbf{x}_c = \mathbf{x}_{i-1} + \mathbf{E}$

$p_c = p(\mathbf{x}_c | \mathbf{A})$

if $(p_c/p_{i-1} > U(0,1))$

$\mathbf{x}_i = \mathbf{x}_c; p_i = p_c$

else

$\mathbf{x}_i = \mathbf{x}_{i-1}; p_i = p_{i-1}$

28

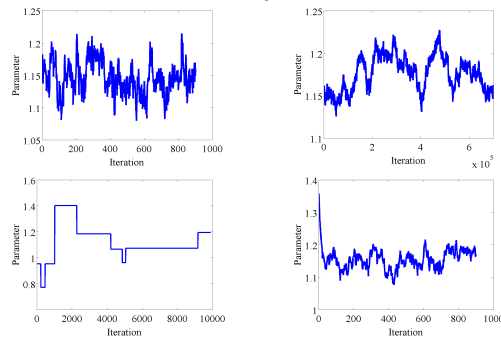
Algorithm parameters

- Burn-in
 - Discard the first B samples to allow the chain to converge on the right distribution from the starting point.
- Sampling interval
 - Only keep every I-th sample to ensure independence of consecutive samples
- Number of samples
 - Must be enough to define the statistic of interest precisely.
- Perturbation size
 - Must reflect parameter scale
 - Rule of thumb: aim for acceptance rate of 20-50%.

Perturbations

- Must have detailed balance: probability of transition from \mathbf{x} to \mathbf{x}' is the same as from \mathbf{x}' to \mathbf{x} .
- Gaussian perturbations for continuous variables are common and practical.
- What about other surfaces, eg sphere?

Common problems



MCMC

- Always check the chains
- Run initially for a very long time to gauge behaviour.
- Tune perturbation sizes for each parameter
- Check convergence of your statistic of interest.
- Lots of variations:
 - Reversible jump
 - Metropolis versus Gibbs sampling.

Tractography again

- Parker et al JMRI 2003 (Parametric bootstrap)
- Behrens et al MRM 2004 (MCMC)
- Lazar et al NeuroImage 2005 (Repetition bootstrap)
- Friman et al TMI 2006 (Laplace's method... sort of).
- Jones et al TMI 2008 (Wild bootstrap)
- Is the distribution $p(\mathbf{x}|\mathbf{A})$ really what we want?

Summary

- Uncertainty of parameter estimates is important to know
 - For reliability of inference
 - For post-processing
- Laplace method
- Resampling methods: bootstrap
- MCMC