Computational modeling for biomedical imaging

Daniel C. Alexander
Centre for Medical Image Computing
and Department of Computer Science,
University College London,
Gower Street, London WC1E 6BT, UK.

January 26, 2015

Correspondence: D.Alexander@ucl.ac.uk.

Contents

1 Introduction

NOTE 0: Box on Gauss or Galileo. Who did this kind of indirect measurement first?

Indirect measurement aims to measure a quantity that we cannot observe directly, but for which we can measure some other quantity that is sensitive to it. That measurement enables us to infer the quantity we are interested in. To make an inference in this way requires a model. The model encapsulates the influence of the quantity of interest on the measurement we make, as well as that of any other effects that influence it. Once we have the model, we can estimate the quantity of interest by fitting the model to the observed data, i.e. we seek the value of the quantity, as well as the values of other influential quantities, that best explain the measurements given the model.

Let's take an example from astronomy to illustrate. The discovery of planets around stars other than our own sun, so-called exoplanets, has generated a lot of excitement since its inception in the mid-1990s. We do not have telescopes anywhere near powerful enough to see these planets directly. However, we can infer their presence by monitoring fluctuations in light arriving from the host star. One sign is a dimming effect from eclipses as the planet moves in front of the star from the Earth's viewpoint. We can observe these transits directly in our own solar system; figure 1 shows a composite image captured from the transit of venus across the sun as seen from Earth in 2012. While the planet eclipses the sun, it reduces the total surface area from which Earth receives light, so the overall light intensity reduces. Other stars are so far away, that the light source is effectively a single point. Thus we cannot observe the transit directly, but do see the dip in intensity during a planetary eclipse.

Figure 2 shows a schematic of how the light intensity varies as a planet orbits a star. The duration and size of the intensity reduction during the eclipse depend on properties of the planet that are not directly observable, such as its size, orbiting distance, and speed. Various other effects also reveal the presence and nature of exoplanets. For example, the position of the star shows slight variations as it moves under the gravitational influence of orbiting planets. We can *observe* the time varying intensity and position of a star. We can *infer* the presence of a planet and make indirect measurements of its size, orbiting distance, orbit eccentricity, and various other features by fitting a model to find the values of those quantities (*parameter estimates*) that explain the observations.

Another example is weather forecasting. We cannot measure today directly what the weather will be like in London tomorrow, but we can make inferences about it from the weather measurements today in London and its vicinity. Modern weather prediction systems feed measurements of localised temperatures, atmospheric pressure, wind speeds etc, into computer simulations of fluid dynamics to predict how things will evolve over the next day or so. As we all know, this works quite well in the short term, and the predictions of the next day's weather are usually quite accurate. However, incomplete understanding of atmospheric dynamics and inaccuracies in the models driving computer simulations mean that prediction accuracy degenerates rapidly with time, at least for complex weather systems; we all know not to trust the end of the five-day weather forecast here in the UK!

The emphasis of this course is on using this kind of model-based inference in imaging applications. Quantitative imaging has become increasingly popular and available in recent years, as imaging hardware has become more powerful,

but relies on sophisticated statistical and computational tools. Many such tools are readily available in standard software packages, such as Matlab, and ostensibly straightforward to run. However, lack of understanding of the tools can lead to poor results and mislead inferences. Here, we aim to provide a conceptual understanding of these tools, practical knowledge of when and how to use them, and what pitfalls to avoid. The tools themselves are not specific to imaging and have much wider application, but here we frame their description within imaging applications and tailor the content, descriptions and examples for the imaging scientist. The focus of the material is on tools for using models rather than devising the models themselves. The latter has no magic recipe and comes rather from deep understanding of the observed objects and observing devices, mixed with a pinch of scientific genius. We make no attempt to teach this, but focus instead on exploiting models once we have them, for estimating parameters, our confidence in those estimates, and for tailoring models and measurements to provide the best possible information. Of course, an understanding of these concepts is of great benefit to designing models themselves, but that requires further domain specific knowledge and intuition.

1.1 Basic terminology

A model predicts the signal we obtain from a system given the properties of the system. A mathematical model expresses the relationship between the properties of the system and the observable signal as set of equations. More complex systems require computational models that use algorithms to predict the signal.

For our exoplanet example, we might write a simple mathematical model of the form

$$I(t) = \frac{L}{d_1^2} - f_1(R, d_2, \omega, \phi; t) \tag{1}$$

and

$$\mathbf{r}(t) = \mathbf{r}_0 - f_2(R, d_2, \omega, \phi; t) \tag{2}$$

where $I(t)$ and $\mathbf{r}(t)$ are the light intensity and source position, respectively, at time t, L is the luminance of the star, d_1 is its distance from Earth, R is the planet radius, d_2 its distance from the host star, ω and ϕ are the frequency and phase of the orbit, and f_1 and f_2 are mathematical functions the precise form of which we shall not be too concerned about here, but see (Nature, vol. 513 18 September 2014) for a summary of the state of the art.

For a simple solar system with a single planet, even two or three planets, the relationship between the planetary variables and the star's position and intensity is straightforward enough to write down as simple mathematical formulae. However, imagine a much more complex solar system with, say, 9 planets each with its own moons, various comets and asteroid belts. Mathematical equations governing such systems rapidly become too complex to be useful. In the modern age, however, computers are easily powerful enough to simulate the mechanics of such systems to predict the position of each interacting body at any given time and thus to predict observable features of the system such as the intensity and apparent source location from Earth. Thus, computational models are possible for very complex systems. In practice, of course, our observations lack the sensitivity to say anything about systems with more than two or three planets, so simple mathematical models are often sufficient. In weather prediction, on the other hand, computational models are standard, because the interaction of different effects is too complex to

capture in simple mathematical form.

Parameters are variables of the system that affect the observable signal. Some parameters are known and under the control of the experimenter, such as the values of the time variable t in the exoplanet example at which each measurement is acquired. These are called independent variables. Others are not under our control and are properties of the system under investigation. These are dependent variables. **NOTE 1: Independent variables answer the question"What do I change?". Dependent variables answer the question"What do I observe?".** Some dependent variables may be known, such as the distance, d_1, from Earth to the host star. Others, such as the planetary features, are unknown. At least some of the unknown parameters are the interesting quantities that we aim to estimate, such as the size of the planet and its distance from the star. Others may be uninteresting in themselves, but necessary to predict the signal accurately and thus recover an estimate of the interesting parameters. For example, the phase parameter ϕ in the model above is not particularly interesting in terms of discovering our universe, but essential to complete the relationship between the interesting parameters and the observable signal. Such parameters are often called *nuisance variables*. **NOTE 2: Wikipedia definition: a random variable that is fundamental to the probabilistic model, but that is of no particular interest in itself.**

Data are the measurements from the device(s) observing the system. The system emits a signal, which the device measures. Signal and measurement are important to distinguish. The measurement approximates the signal, but is distinct because of the influence of noise.

Noise is a general term for any influence on the data that is not explained

by the model. **NOTE 3: Wikipedia definition: colloquialism for recognized amounts of unexplained variation in a sample.** Noise comes in various forms:

- Measurement noise. All measurements are affected by random fluctuations arising from imperfections in the measurement device. Consider our telescope viewing the star with an exoplanet. Most likely it uses a digital camera to measure the light intensity; the simplest form is a charge-coupled device (CCD). The CCD consists of a sensor that produces electrical charge when photons hit it. The sensor is attached to a capacitor, that stores the charge like a bucket. After a predetermined exposure time, the devise counts the amount of charge in the capacitor and that provides the measure of light intensity. Thermal energy in the system also creates charge that fluctuates randomly. This adds random variation to the measurement. Although the influence of this kind of noise is unpredictable on any individual measurement, we can often come up with a fairly precise statistical description.

- Quantization and truncation error. Measurements are often digitised. For example, our intensity measurement from the CCD typically goes through an analog to digital converter (ADC) that turns the capacitor charge into, say, a 16-bit integer stored in a computer. This inevitably involves some loss of precision and can cause close to identical signals to produce a different measurement. However, so long as the dynamic range of the analog to digital conversion is chosen appropriately, this source of error is usually negligible. Truncation error is a more serious problem. For example, capacitors in CCDs have finite capacity and can fill up placing a limit on the intensity measurement; we cannot

say anything about how much larger the intensity really was than the maximum.

- Catastrophic failure. Something goes completely wrong in the experiment and the measurement does not reflect the signal at all. Possible ways in this might happen in our exoplanet example include: a bird flying in front of the telescope so the device does not observe the system of interest at all. These kinds of effect cause what we often refer to as outliers.

- Modelling error. The model does not predict the signal, because it ignores an important effect. For example, our model of planetary motion above assumes circular rather than elliptical orbits. It also assumes a step change from the planet not being in front of the star to being fully in front, whereas actually there is a gradual transition.

Now we can break the idea of a model down into several common components:

- A signal model. This describes the system that generates a signal. It has parameters

$$\mathbf{x} = (x_1, \cdots, x_N)^T \tag{3}$$

at least some of which are unknown and all of which are *dependent variables*. In our exoplanet example, the signal model is Eqs. 1 and 2. It describes the signal we receive on Earth in terms of variables of the remote solar system, which are the x_i.

- A device or measurement model. This describes how the device turns an input signal into a measurement. The device has several parameters

$$\mathbf{y} = (y_1, \cdots, y_M)^T \tag{4}$$

that usually are known and controllable, so *independent variables*. In our running example, these are settings of the telescope, such as exposure time, gain settings of the ADC, interval between successive measurements.

- A noise model. This describes, usually statistically, the random processes that affect the measurement of the signal. It has parameters

$$\mathbf{z} = (z_1, \cdots, z_L)^T \tag{5}$$

which may be known or unknown. In our example, these might describe the random fluctuations in the telescope output from thermal energy, which we might describe statistically by their standard deviation. Various simple noise models are common. For example, in an additive noise model the measurement

$$A = S(\mathbf{x}, \mathbf{y}) + \eta(\mathbf{z}) \tag{6}$$

where S is the signal and η is the noise term, which is independent of the signal. Alternatively, in a multiplicative noise model,

$$A = S(\mathbf{x}, \mathbf{y})(1 + \eta(\mathbf{z})), \tag{7}$$

so the level of noise depends linearly on the signal.

The full model then predicts the signal deterministically via the physical and device models, but the measurements statistically via the noise model. Figure 3 illustrates schematically how the model components combine.

1.2 What can we do with models?

Models are used in various ways in different contexts. Common modes of usage are:

- To learn about the world. Is it possible that the existence of an exoplanet could explain the variation in intensity of light that we observe from stars? Is there an exoplanet there at all? These were the first questions to be asked in exoplanet research. The answer is fairly firmly established as "yes" these days and the emphasis has switched to the next item.

- To estimate interesting features (parameters) of the system. In the exoplanet example, we might ask: is the planet Earth like; might it support life? Currently, we aim to estimate parameters like the size and orbiting distance, which is a first step towards answering this kind of question.

- To predict future events and measurements. E.g. next year on February the 14th at 8:37 the next eclipse of planet X83957 of star 48582834 will start. **NOTE 4: This is made up... check Nature 513 for a real example...**

Our focus in this text is on applications of modelling and parameter estimation in biomedical imaging, although, as the other examples above illustrate, the basic techniques have much broader application.

Figure 1: Composite image of the venus transit of the sun in 2012; from Pasachoff et al Nature 485, 2012.

Figure 2: Schematic diagram of star light intensity variation resulting from planetary eclipses; from Wikipedia.

Figure 3: Shows how the components of a typical model supporting indirect measurement fit together.

2 Estimation

NOTE 5: Fisher: invented maximum likelihood estimation as an undergraduate.

We estimate parameters by *fitting* a model to some data. The fitting procedure searches for the combination of unknown parameter values for which the model best predicts the measured data. We minimise, with respect to the parameter values, some measure of distance between the actual measurements and those that our model predicts. The measure of distance encodes the statistical fluctuations in our measurements that the noise model predicts as well as any prior knowledge we might have about the values of the unknown parameters.

Optimization algorithms are numerical techniques for locating the minima (or maxima) of functions. They seek

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} f(\mathbf{x}), \tag{8}$$

where f is the objective function, and $\mathbf{x}$ is the vector of parameters. Optimization is the workhorse of parameter estimation, where it finds the minima of objective functions encoding the distance measures we refer to above.

In this section, we discuss various distance measures we might use for parameter estimation and various optimisation algorithms we might use to minimise them. We finish by introducing a real-world parameter estimation problem from biomedical imaging, which serves as an example of the ideas in this chapter as well as the rest of the document.

2.1 Distance and similarity

This section introduces and motivates various common choices of distance measure or objection function for model fitting and parameter estimation.

2.1.1 Least squares

A common objective function for fitting models to data is the sum of square differences:

$$f(\mathbf{x}) = \sum_{k=1}^{K} (A_i - S(\mathbf{x}, \mathbf{y_k}))^2 \tag{9}$$

where K is the number of measurements, each of which has unique device settings $\mathbf{y}_k$, A_k is the k-th measurement, and $S(\mathbf{x}, \mathbf{y_k})$ is the prediction of the k-th measurement from the model with parameter settings $\mathbf{x}$. The set of parameters $\tilde{\mathbf{x}}$ that minimise the sum of square differences, $f(\mathbf{x})$ in Eq. 9, is the *least-squares estimate* of $\mathbf{x}$.

Figure 4 compares measurements of a child's height at four successive birthdays against a linear model fitted to the data points. The least squares fit is the line that minimises the sum of squared vertical distances between itself and the data points. The fitted model provides a prediction of the heights at unobserved ages, such as in between birthdays and at the next birthday.

2.1.2 Maximum likelihood estimation

The maximum likelihood principle says that we should find the estimate that maximises the likelihood of the data we observe given a statistical model of

the noise. Mathematically, we seek the solution

$$\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{A}|\mathbf{x}), \tag{10}$$

where $\mathbf{A}$ is the vector of measurements. If the noise on each measurement is independent, then

$$\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}} \prod_{k=1}^{K} p(A_k|\mathbf{x}). \tag{11}$$

Since the log function is monotonic increasing, the log of the product in Eq. 11 has the same maximum, so that

$$\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}} \sum_{k=1}^{K} \log p(A_k|\mathbf{x}). \tag{12}$$

A pleasing statistical interpretation of the least squares solution is that it is the maximum likelihood estimate of the parameters under the assumption of unbiased independent identically distributed Gaussian noise. That means that the noise term $\eta(\mathbf{z})$ in each measurement, as written in Eq. 6, follows a zero-mean Gaussian distribution, i.e. $\eta(\mathbf{z}) = A_k - S(\mathbf{x}, \mathbf{y_k}) \sim N(0, \sigma)$. The vector, $\mathbf{z}$, of noise parameters has a single entry, σ, and the noise components of two separate measurements give no information about one another. With this Gaussian noise model, by definition

$$p(A_k|\mathbf{x}) = (2\pi\sigma^2)^{-1} \exp(-\frac{(A_k - S(\mathbf{x}, \mathbf{y_k}))^2}{2\sigma^2}) \tag{13}$$

so

$$\log p(A_k|\mathbf{x}) = -\log(2\pi\sigma^2) - \frac{(A_k - S(\mathbf{x}, \mathbf{y_k}))^2}{2\sigma^2}. \tag{14}$$

We can see that the maximum likelihood estimate of $\mathbf{x}$ is the least-squares estimate by substituting Eq. 14 into Eq. 12. We can drop the constant terms, $\log(2\pi\sigma^2)$, as they are independent of $\mathbf{x}$ and do not affect the location of the

maximum. Similarly, we can drop the scaling factor $(2\sigma^2)^{-1}$ to obtain

$$\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}} \sum_{k=1}^{K} -(A_k - S(\mathbf{x}, \mathbf{y_k}))^2, \tag{15}$$

or equivalently

$$\tilde{\mathbf{x}} = \arg\min_{\mathbf{x}} \sum_{k=1}^{K} (A_k - S(\mathbf{x}, \mathbf{y_k}))^2, \tag{16}$$

which is the least squares estimate.

We can relax the assumption that the noise distribution is identical on each measurement slightly by allowing the standard deviation of the Gaussian noise to vary among measurements, which leads to the weighted least squares objective function

$$f(\mathbf{x}) = \sum_{k=1}^{K} \frac{(A_k - S(\mathbf{x}, \mathbf{y_k}))^2}{2\sigma_k^2}. \tag{17}$$

2.1.3 Robust statistics

Various alternative objective functions are common for parameter estimation. One limitation of the sum of square differences measure is that it is sensitive to outliers, which are freak measurements with large error arising typically from catastrophic failures of the measurement process; see figure 5.

Other objective functions reduce the weight on outliers so produce less sensitive fitting results. One simple choice is the L1 norm

$$f(\mathbf{x}) = \sum_{k=1}^{K} |A_k - S(\mathbf{x}, \mathbf{y_k})|. \tag{18}$$

This is one example of a class of so-called robust statistics; robust to outliers. One specific class of robust statistics are M-estimators, which have the

general form

$$f(\mathbf{x}) = \sum_{k=1}^{K} \rho(A_k - S(\mathbf{x}, \mathbf{y_k})). \tag{19}$$

Thus both the least-squares and the L1-norm objective functions are M-estimators; for the former, $\rho(r) = |r|$ (the absolute value), and for the latter $\rho(r) = r^2$. Other choices include Winsoring, which uses $\rho(r) = r^2$ close to $r = 0$, but switches to $\rho(r) = |r|$ for r greater than some threshold to down weight the influence of outliers. Tukey's biweight function reaches an asymptote as r becomes large, which down weights outlier influence even further. Figure 6 illustrates various common M-estimators.

We can obtain some robust statistics, or M-estimators, through alternative choices of the noise model in the maximum likelihood principle. The L1-norm estimator in Eq. 18, for example, corresponds to a Laplacian noise model, where the noise distribution is a Laplacian distribution, which is a double-sided exponential:

$$p(A_k|\mathbf{x}) = (2b)^{-1} \exp(-\frac{|A_k - S(\mathbf{x}, \mathbf{y_k})|}{b}). \tag{20}$$

A common robust choice of noise model is Student's t-distribution, which looks a bit like a Gaussian distribution, but with heavier tales so that outliers are more likely, have less negative log likelihood and thus smaller influence on the objective function and fitted parameters. Figure 7 compares various distributions that provide robustness to outliers with the Gaussian.

2.1.4 Bayesian estimation

Maximum likelihood estimation maximises the likelihood, $p(\mathbf{A}|\mathbf{x})$, of the data given the model parameters. A more intuitive quantity to maximise is the

posterior likelihood, $p(\mathbf{x}|\mathbf{A})$, of the parameter estimates given the data to obtain the maximum a-posteriori (MAP) estimate. We can relate the two using a standard tool of conditional probability, Bayes rule:

$$p(\mathbf{x}|\mathbf{A}) = \frac{p(\mathbf{A}|\mathbf{x})p(\mathbf{x})}{p(\mathbf{A})}. \tag{21}$$

Given a noise model, we can compute $p(\mathbf{A}|\mathbf{x})$ easily. The difficulty in computing $p(\mathbf{x}|\mathbf{A})$ is in computing the other two terms: $p(\mathbf{x})$, the prior on the parameter settings, and $p(\mathbf{A})$ the prior on the data, sometimes called the *evidence*.

The prior on the parameter settings is our belief in each possible combination of parameter values before any measurements are made. It has to be set using some kind of intuition, which gives rise to the common objection to Bayesian approaches: you can cheat by setting this to something that gives you the answer you want. However, we can also set the prior to be very uninformative so that it rules out physically unrealistic combinations, but treats everything else as equally likely a-priori.

The term $p(\mathbf{A})$ is the overall likelihood of the measurements. We can write this

$$p(\mathbf{A}) = \int p(\mathbf{A}|\mathbf{x})p(\mathbf{x})d\mathbf{x}. \tag{22}$$

In many practical situations, the integral in Eq. 22 is intractable. However, the term does not depend on $\mathbf{x}$, so proves unimportant for parameter estimation. For maximising $p(\mathbf{x}|\mathbf{A})$ we can simply maximise the numerator and forget about the denominator.

Thus the MAP estimate is

$$\tilde{\mathbf{x}} = \arg\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{A}) = \arg\max_{\mathbf{x}} (\log p(\mathbf{A}|\mathbf{x}) + \log p(\mathbf{x})). \tag{23}$$

Here is a simple example to illustrate the difference between maximum likelihood and maximum a-posteriori estimation. Suppose I tell you that I have measured a man's height and obtained a figure of 1.955 m. What's our best guess as to his actual height? Let's suppose I made the measurement by counting handspans from his feet to his head. My handspans are about 23 cm and I made it 8 and a half hand spans, hence the figure 1.955 m. The variance is relatively high on this measurement, although probably not biased upwards or downwards, so we might take as a noise model a zero-mean Gaussian distribution with standard deviation about half a hand-span; let's say 10 cm. Since the noise is unbiased, the maximum likelihood estimate is indeed 1.955 m. However, the MAP estimate takes into account that the mean height is considerably lower than 1.955 m. Men's heights are approximately Gaussian distributed with mean around 1.7 m and standard deviation around 5 cm. If we take that distribution as our prior on this man's height and combine with our data, we find that the posterior distribution is proportional to $N(1.955, 0.1)N(1.7, 0.05)$, which has a maximum around 1.75 m. Thus, we end up with a MAP estimate closer to the average man's height than our measurement, because heights lower than 1.955 m have much higher prior probability than heights greater than 1.955 m and we have more confidence in our prior than in our measurement. Similarly, if the height measurement were much lower than the average, the MAP estimate would be higher than the measurement. However, now suppose we measure the height properly using a ruler so that the standard deviation on the measurement reduces to something like 1 cm; how does this affect our MAP estimate? Figure 8 illustrates this example.

2.2 Optimisation

Once we have decided what function to minimise to obtain parameter estimates, we need to decide how to minimise it. This we do through an optimisation procedure the choice of which depends on the model and the form of the objective function.

2.2.1 Linear models

Some particularly simple and convenient minimisation algorithms are available for linear models, which have the form

$$S(\mathbf{x}, \mathbf{y}) = \sum_{i=1}^{N} g_i(\mathbf{y}) x_i. \tag{24}$$

The model is linear in the parameters, $x_i, i = 1, \cdots, N$, but not necessarily the independent variables, $\mathbf{y}$, so the functions $g_i, i = 1, \cdots, N$ can have any form.

Suppose we acquire a set of measurements and list them in the vector $\mathbf{A} = (A(\mathbf{y}_1), \cdots, A(\mathbf{y}_K))^T$. To find parameters that best fit these measurements, we can solve the matrix equation

$$\mathbf{A} = G\mathbf{x}, \tag{25}$$

where the ik-th entry of the design matrix G, $G_{ik} = g_i(\mathbf{y}_k)$. So long as G has rank N, we can solve for $\mathbf{x}$ by premultiplying each side of the matrix equation by G^T,

$$G^T\mathbf{A} = G^T G\mathbf{x}, \tag{26}$$

and then the inverse of the symmetric matrix $G^T G$ to obtain

$$\mathbf{x} = (G^T G)^{-1} G^T \mathbf{A}. \tag{27}$$

The matrix $(G^TG)^{-1}G^T$ is the Moore-Penrose pseudo-inverse of G. Note that, in general, G is not directly invertible, as it may not be square, but G^TG is always invertible so long as G has rank N. In matlab, `pinv(G)` computes the Moore-Penrose pseudo-inverse. Solving linear systems of equations is so common that Matlab also provides the short-cut `x = G\A`, which solves Eq. 25 directly, although it may not use the Moore-Penrose pseudo-inverse as alternatives exist that are sometimes more numerically stable.

The estimate of $\mathbf{x}$ that we obtain from Eq. 27 is in fact the least squares solution. To see this, let's write the sum of square differences objective function

$$f(\mathbf{x}) = \sum_{k=1}^{K} r_k^2, \tag{28}$$

where

$$r_k = A_k - S_k(\mathbf{x}, \mathbf{y}_k) = A_k - \sum_{i=1}^{N} G_{ik} x_i \tag{29}$$

is the residual error of the k-th measurement. At the minimum, the derivative of f with respect to each parameter is zero, i.e.

$$\frac{\partial f}{\partial x_i} = 2 \sum_{k=1}^{K} r_k \frac{\partial r_k}{\partial x_i} = 0 \tag{30}$$

for each $i = 1, \cdots, N$. From Eq. 29,

$$\frac{\partial r_k}{\partial x_i} = -G_{ik}. \tag{31}$$

Substituting into Eq. 30,

$$2 \sum_{k=1}^{K} \left((-G_{ik})(A_k - \sum_{j=1}^{K} G_{jk} x_j) \right) = 0, \tag{32}$$

and rearranging

$$\sum_{k=1}^{K} \sum_{j=1}^{K} G_{ik} G_{jk} x_j = \sum_{k=1}^{K} G_{ik} A_k, \tag{33}$$

which, combining over all $i = 1, \cdots, N$, is exactly the matrix equation in Eq. 26 so Eq. 27 must be the least squares solution.

Consider the child's heights example from figure 4 with the linear model

$$h(a) = ma + c \tag{34}$$

where h is the height at age a years and m and c are the unknown parameters of the model. The set of parameters, $\mathbf{x}$, is $\{m, c\}$ and the set of known settings, $\mathbf{y}$, is $\{a\}$. Let's assume we have four height measurements $\mathbf{h} = (h_1, h_2, h_3, h_4)$ at each of ages a_1, a_2, a_3, a_4, so we have four equations, $h_i = ma_i + c, i = 1 \cdots 4$, in our two unknowns. With reference to Eq. 24, $g_1(\mathbf{y}) = a$ and $g_2(\mathbf{y}) = 1$. Thus, we can construct the matrix equation $\mathbf{h} = G\mathbf{x}$, where

$$G = \begin{pmatrix} a_1 & 1 \\ a_2 & 1 \\ a_3 & 1 \\ a_4 & 1 \end{pmatrix}, \tag{35}$$

and solve using Eq. 27.

The alternative model

$$h(a) = pa^2 + ma + c \tag{36}$$

is still linear despite the quadratic term in a. The model is linear in its three unknown parameters, $\mathbf{x} = \{p, a, c\}$, even though the functions, $g_1(\mathbf{y}) = a^2$, $g_2(\mathbf{y}) = a$ and $g_3(\mathbf{y}) = 1$, are no longer linear. The design matrix becomes

$$G = \begin{pmatrix} a_1^2 & a_1 & 1 \\ a_2^2 & a_2 & 1 \\ a_3^2 & a_3 & 1 \\ a_4^2 & a_4 & 1 \end{pmatrix}, \tag{37}$$

and we solve for $\mathbf{x}$ in exactly the same way using Eq. 27. However, the model

$$h(a) = p\exp(ma) + c \tag{38}$$

is not linear in the three parameters so we cannot estimate them in this way. Figure 9 shows fits of each model to some example data to illustrate the differences.

2.2.2 Weighted linear least-squares

The linear least-squares solution in Eq. 27 assumes that the noise on each measurement is identically distributed, i.e. the standard deviation of the Gaussian noise on each measurement is equal. The linear estimation procedure extends in a straightforward way to accommodate the situation where the standard deviation is different for each measurement, i.e. to minimise the weighted sum of squared difference objective function in Eq. 17. The weighted linear least squares solution is

$$\mathbf{x} = (G^T W G)^{-1} G^T W \mathbf{A}. \tag{39}$$

where W is the diagonal $K \times K$ matrix with $W_{kk} = \sigma_k^{-2}, k = 1, \cdots, K$.

NOTE 6: Other weighted least-squares solutions?

NOTE 7: Sections to add here: quadratic programming; methods for minimising the L1 norm - LASSO; basis pursuit etc.

2.2.3 Non-linear optimisation

In general, models are not linear in the parameters and the noise is not Gaussian distributed, so we need alternative ways to minimise objective functions

to obtain parameter estimates. This requires non-linear optimisation techniques. Non-linear optimisation is a huge field of research and I shall not attempt to cover it in detail here, but simply give an overview of the standard techniques as well as pros and cons of the different approaches.

Non-linear optimisation algorithms iteratively improve parameter estimates until they reach a point where they can find no further improvement in the objective function. They seek a sequence $\mathbf{x}_0, \mathbf{x}_1, \cdots$ that converges to a minimum (or maximum) point $\tilde{\mathbf{x}}$ where $\nabla f(\tilde{\mathbf{x}}) = 0$. Different techniques locate subsequent elements of the sequence in different ways. One standard strategy is gradient descent, which takes each step in the most downhill direction, i.e.

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \gamma \nabla f(\mathbf{x}_i), \tag{40}$$

where γ is some fixed step length parameter, until it can no longer find a downhill step. Newton methods further exploit the curvature of the objective function by using its Hessian matrix (second derivative):

$$\mathbf{x}_{i+1} = \mathbf{x}_i + \gamma H^{-1}(\mathbf{x}_i) \nabla f(\mathbf{x}_i). \tag{41}$$

The latter often converge much more quickly, particularly if the objective function has long thin valleys leading to the minimum. Figure 10 illustrates this. Gradient descent tends to zigzag along the valley, whereas the Newton method makes steps more directly towards the minimum point. Whereas gradient descent approximates the function locally as linear and goes down hill, the Newton method approximates it as a quadratic and takes a step towards the minimum of the quadratic. The latter usually takes us closer to the minimum, as, by definition, the objective function is non-linear. However, the disadvantage of the Newton method is that it requires computation of the Hessian matrix at each step, which can be complex to implement and

computationally expensive.

Several variations of these fundamental methods are widely used:

- Quasi-Newton methods use the principle of the Newton method but avoid direct computation of the Hessian matrix by approximating it numerically from successive iterations of the minimisation.
- Gauss-Newton methods are specifically for non-linear least-squares problems, where we can get a good estimate of the Hessian matrix from the Jacobian (the first derivative matrix). Specifically, it replaces H with $2J^TJ$ where J is the Jacobian matrix. **NOTE 8: Check.**
- The Levenberg-Marquardt algorithm iterates between gradient descent when the sequence is reducing slowly and Gauss-Newton when it is reducing quickly. This improves the robustness of the search and enables it to find the global minimum from a broader range of starting points. **NOTE 9: Why?**

The matlab function `fminunc` is the generic unconstrained non-linear optimisation algorithm and the function `lsqnonlin` is similar but specifically for least squares problems. Both use versions of the algorithms above and allow you to choose and compare various choices and settings.

Other deterministic non-linear optimisation techniques include

- The Simplex method (`fminsearch` in matlab), which iteratively moves the vertices of a polygon containing the minimum towards the minimum until they are within a predefined distance of one another. It does not require derivates so is very simple to employ.

- Powell's method, which is a gradient descent technique that also does not require computation of the derivatives of the objective function.

While methods that do not require derivatives make life easier for the user, insofar as they are straightforward to use, they are often slower and less reliable than search strategies that use derivatives for similar reasons to those illustrated in figure 10.

2.2.4 Constrained optimisation

Often we need to impose constraints on parameter estimates to ensure physically realistic solutions. There are several ways to force optimisation algorithms to obey such constraints.

- Transformation method. Often we can transform our parameters so that the problem becomes an unconstrained optimisation problem. For example, if we know that one of our parameters, say x_1 must be positive, we can write it as the square of another variable $x_1 = \alpha_1^2$ where α_1 can take any real value. We replace x_1 by α_1^2 in the model; use non-linear optimisation to find the best estimates of α_1 and $x_2, \cdots, x_N$, then set $\tilde{x}_1 = \tilde{\alpha_1}^2$. To constrain x_2 to the range $[0, 1]$, we might set $x_2 = \sin^2 \alpha_2$ or $x_2 = \exp(-\alpha_2^2)$.

- Soft or penalised constraints. Add a term to the objective function that penalises violations of known constraints. For example, to encourage x_1 to be positive, we might find the minimum of the function

$$f' = f + A.H(-x) \tag{42}$$

where f is the original unconstrained objective function, A is some large constant and H is the Heaviside step function:

$$H(x) = 0, x < 0; H(x) = 1, x > 0. \tag{43}$$

Often stability is better if the penalising term is smooth, for example

$$f' = f + A.x^2 H(-x). \tag{44}$$

- Lagrange multipliers **NOTE 10: Any different to soft constraint?** and active set methods. These techniques can be incorporated into non-linear optimisation algorithms to enforce constraints on the parameters. A range of generic techniques exist and are widely used; see the `fmincon` function in matlab and its help page for a description and implementation. Although they slow each iteration, explicit constraints can produce more reliable performance and provide faster overall convergence than the transformation method, which can make the Hessian of the objective function ill-conditioned causing numerical instability.

Recall our non-linear model for the child's height

$$h(a) = p \exp(ma) + c. \tag{45}$$

Some parameter combinations are unrealistic, for example we could imagine a solution with large and negative m, large and negative p and large and positive c, which fits the data reasonably well; see figure 11. Such a solution is unrealistic however, as it has negative height at small ages. A more realistic solution has all three parameters small and positive. We might thus constrain the optimisation to find this kind of solution by forcing all three parameters

to be positive. The transformation method can achieve this by setting

$$h(a) = p_1^2 \exp(p_2^2 a) + p_3^2. \tag{46}$$

We fit to the data to estimate p_1, p_2, p_3 and then set $\tilde{p} = \tilde{p}_1^2$, $\tilde{m} = \tilde{p}_2^2$, and $\tilde{c} = \tilde{p}_3^2$.

2.2.5 Local minima

A ubiquitous problem in non-linear optimisation and parameter estimation is that of local minima. When a function has more than one minimum point, all the techniques discussed above converge to the minimum closest to their starting point, which may not be the global minimum; figure 12 illustrates the problem. In this one dimensional example, if we start anywhere to the right of the maximum point near the centre and go downhill, we find the global minimum, but if we start the other side of the maximum, we always end up in the local minimum. Complex functions can have many local minima. Consider the contour map in figure 13, which is more representative of the kind of objective function we often face in real-world optimisation problems. If a hiker starts in the village called Stowe and walks uphill at the sharpest angle possible until he cannot go uphill any further, where does he end up? Quite possibly only at the local hillock marked with an X on the figure. For the constrained optimisation problem limiting our hiker to this map the global maximum is Tabor Hill on the bottom right. To locate the maximum without the benefit of the map, the hiker would need to start in lots of different places and repeat the uphill climb. This problem is only two dimensional whereas real-world optimisation problems often have many more dimensions and searching for the global maxima in high dimensional functions with this degree of complexity can be very challenging.

2.2.6 Stochastic optimisation

In practice, most fitting problems suffer from local minimum problems to a greater or lesser extent. Non-deterministic, stochastic search procedures can improve our chances of finding the global optimum. Various strategies are available:

- Repeated gradient descent from different starting locations. A common strategy is to run deterministic non-linear algorithms from lots of different starting points; figure 14. Often the set of starting points are chosen randomly within sensible bounds, but they may also be defined on a grid or some other regular pattern within a sensible range. The more starting points we use, the more likely it is that at least one run finds the global minimum. However, for complex objective functions we can never be sure that any of the runs have done so. One must be careful to choose the range of perturbations of the starting points appropriately to ensure that, first, they are large enough to escape from the capture range of the local minimum, and second, they are small enough have a good chance of starting in the capture range of the global minimum (assuming it is somewhere in the vicinity of the initial starting point). Without prior knowledge of the landscape of the objective function, we can never be sure that we have found the global minimum. One helpful practical approach is to look at the histogram of final objective function values; if the smallest value is unique, it is still very likely a local minimum; if a good proportion of the searches end at the smallest value, we might have more confidence that it is the global minimum.

- Simulated annealing is a stochastic search strategy. **NOTE 11: Algorithm to be included.** The algorithm searches the space of possible solutions probabilistically in such a way that moves from less likely solutions to more likely solutions happen more often than vice versa, but both uphill and downhill moves are possible, which allows the search to escape from local minima. The basic search strategy is Metropolis-Hastings sampling, which we will come across later when we discuss Markov Chain Monte Carlo (MCMC), but during the procedure the step length of the perturbations decreases from very large early on, to ensure a broad search for the capture range of the global optimum, to very small at the end to refine the search to the precise optimum. The search is guaranteed in the limit to converge to the global minimum of any objective function. However, that limit can be very very long!

- Population-based stochastic search strategies, such as genetic algorithms, differential evolution, and self-organising migratory algorithms, are increasingly popular. **NOTE 12: To include outline algorithm and brief summary of differences.** They work by maintaining a population of candidate solutions to the minimization of the objective function. Each iteration evaluates the function for each candidate solution and combines solutions for the next iteration or generation of the population in such a way that better solutions are more likely to persist to the next generation. Few, if any, of these techniques provably converge to the global minimum, but in practice they tend to find good solutions more quickly that simulated annealing **NOTE 13: check**. The best choice of strategy for propagating populations depends on the problem.

For simple optimisation problems for which the global minimum has a broad capture range, gradient descent type algorithms have a clear advantage over stochastic searches and converge to the minimum much more rapidly and precisely. However, for the majority of interesting problems, local minima abound, and gradient descent must use multiple starting points to have a hope of finding the global minimum. Simulated annealing and population-based searches often then find better solutions in a fixed computation time. The latter also have the advantage that they require no derivatives of the objective function, so work naturally with highly irregular and discontinuous functions. Disadvantages of simulated annealing and population-based searches include poor precision in localisation of minima, since stochastic searches converge much more slowly once within the capture range of the global minimum than a direct gradient descent; often people"finish-off" a stochastic search with a gradient descent to the closest minimum from the best solution returned by the stochastic search. Stochastic search algorithms also have lots of tuneable parameters and often require a lot of manual configuration for specific problems to get them to work efficiently.

Figure 4: Linear model of a child's height as a function of age fitted to four data points.

Figure 5: Linear model of a child's height in the presence of an outlier.

Figure 6: Plots of various m-estimators. **NOTE 15: see Wikipedia robust statistics page for the precise form of Tukey's biweight function**

Figure 7: Laplace and t-distributions compared to a normal.

Figure 8: Likelihood, prior and posterior distribution on height; see text for details.

Figure 9: Fits of various simple models to the child's height data.

Figure 10: Comparison of the convergence properties of gradient descent and Newton's method.

Figure 11: Illustration of local minima in fitting of a non-linear model to the heights data.

Figure 12: The local minimum problem in one dimension.

Figure 13: Local minima in two dimensions.

Figure 14: Illustration of repeated gradient descent on the problem in Fig. 13.

3 Parametric mapping

This section illustrates some of the ideas introduced previously in this section with a specific imaging example. Model-based techniques arise at various stages in the imaging pipeline, but here we focus on a conceptually straightforward application known as parametric mapping. In the simplest form of parametric mapping we fit a model separately in every pixel or voxel (equivalent of a pixel in a 3D image volume) of the image. Normally this requires several raw images each with different settings of the imaging device so that in each voxel we have several measurements, each corresponding to the same point in the imaged sample. We then fit a model to the set of measurements in each pixel/voxel. Fitting the model produces parameter estimates. A parameter map is simply the image we obtain by putting the value of the parameter in each corresponding pixel location.

Figure 15 shows an example where we acquire several MRI images of a monkey brain. In each pixel we fit a model with parameters including cell size and density. The colour overlay is a map of average cell size in each pixel that we obtain by fitting a model to the set of intensities in each pixel; one parameter of the model is cell size. MRI lends itself well to modelling applications, because it is a very flexible imaging technique. The scanner can be tuned in a great many different ways to alter its sensitivity to different physical phenomena. Thus we can obtain combinations of images that support the estimation of the parameters of quite complex models.

3.1 Diffusion MRI

The particular kind of MRI that we will use for this example is a technique called diffusion MRI. In diffusion MRI, the image intensity is sensitive to the dispersion of water molecules arising from Brownian motion. The technique is useful in biomedicine, because the cellular architecture of tissue determines the dispersion pattern. Thus measurements sensitive to the dispersion pattern provide a window on the cellular architecture. Typical diffusion times in diffusion MRI experiments are tens of milliseconds and during that time the root mean squared displacement of water at body temperature is a few tens of microns. Thus the dispersion pattern is sensitive to structures on the order of microns in size, such as cells. These are structures that are much smaller than the actual pixels in a typical MR image, which are usually of order 1mm in each dimension.

Our example application of diffusion MRI is in the brain. The brain contains three broad types of tissue; see figure 16. First, there are fluid-filled regions such as the ventricles, which contain cerebrospinal fluid, which is mostly water. In these regions, water mobility is high as there are no cellular structures to impede the motion of water molecules. Second, white matter is the cabling that links different processing centres within the brain and allows them to communicate with each other and with the rest of the body. It even looks like cabling and consists of bundles of nerve fibres or axons. Water in this environment has high mobility in the direction of the nerve cells because nothing impedes their motion, but has low mobility in the perpendicular directions because the nerve cell walls restrict and impede its motion. Thus the dispersion pattern tends to extend in the direction of the nerve fibres. The third tissue type is grey matter, which has less ordered

architecture, but nevertheless is densely packed with cells that impede water mobility. Thus we see less dispersion than we do in free fluid, but water disperses more or less isotropically unlike the anisotropic dispersion we see arise in white matter from the coherent organisation of the fibrous cells. By observing these different kinds of dispersion pattern, we can make inferences about what kind of tissue resides in each image voxel. In particular, by fitting models to measured signals, we can estimate features of the tissue such as the packing density of cells or the orientation of white matter fibres.

The basic measurement we make in diffusion MRI has five degrees of freedom (five device settings). Figure 17 shows what is called the MRI pulse sequence for the standard diffusion MR image. Think of it as a program that tells the MRI scanner what to do to make the image we want to acquire. The "echo" at the end of the sequence is the signal that we measure and eventually forms the image intensity. It is the net magnetisation of the sample. That magnetisation comes initially from the 90-degree pulse at the start, which aligns the magnetisation of particles (water molecules in our example), referred to as "spins", in the sample. Those magnetizations rotate with frequency depending on the strength of the magnetic field. Since the magnetic field is not spatially homogeneous, they rapidly go out of phase so we lose the net magnetisation. The one-eighty degree pulse at the centre of the pulse sequence negates the phase of each spin. Although the magnetic field is not homogeneous, it is constant in time, so the spins continue to dephase at the same rate and thus come back into alignment at the echo time. We measure the net magnetisation to get our image intensities.

The key feature of the pulse sequence that gives the resulting image sensitivity to the dispersion of water molecules is the pair of pulses of length δ,

one before and one after the 180-degree pulse. The magnetic field gradient is constant during the pulses and the two pulses have the same length and constant gradient. The important variables of the diffusion MRI pulse sequence are thus the diffusion time Δ, which is the time between the starts of the two gradient pulses, the length, δ, of the pulses, and the strength and orientation of the gradient during the pulses; we describe the gradient with a vector $\mathbf{G}$, which is the component of the gradient of the magnetic field in the direction of the field.

3.2 Simple model

We start with about the simplest possible model for the diffusion MRI signal. It enables us to estimate just one useful parameter, which is the diffusivity of water in each image voxel. To do that, we can summarise all the device settings into one single value, which is commonly referred to as the b-value:

$$b = (\gamma\delta\|\mathbf{G}\|)^2(\Delta - \delta/3), \tag{47}$$

where γ is the gyromagnetic ratio, which is a property of the particles creating the signal. We will also make use of the quantity $\mathbf{q} = \gamma\delta\mathbf{G}$; we can write $b = |\mathbf{q}|^2 t$, where t is an effective diffusion time. The b-value quantifies how sensitive the image is to water dispersion. We can control it by varying the strength of magnetic field gradients, $|\mathbf{G}|$, up to a limit imposed by the scanner hardware, the length, δ, of the pulses or their separation, Δ. The model that relates the diffusion MRI signal to the diffusivity d is then

$$S(b) = S(0)\exp(-bd), \tag{48}$$

where $S(0)$ is the signal with the gradient pulses turned off.

A simple way to produce a parametric map of d is to acquire just two images, one with the gradients turned off so that $b = 0$ and another with them turned on so that $b > 0$. Taking logs of Eq. 48, we can solve for d in a straightforward way:

$$d = b^{-1}(\log S(0) - \log S(b)). \tag{49}$$

Thus we simply take logs of the two images, subtract them and divide by the non-zero b-value to obtain a map showing our estimate of d in every pixel. Figure 18 illustrates this.

3.3 Example data set

Diffusion MRI data sets usually have more than the two images we use to map d above. A typical acquisition protocol contains a few images with $b = 0$ and a larger number with equal $b > 0$, but distinct gradient orientation $\hat{\mathbf{q}}$. Figure 19 shows a slice through each image volume comprising such a data set. There are three $b = 0$ images (the bright ones) and thirty images with $b > 0$. Here the b-value for each of those 30 images is $1000\,\mathrm{sm}^{-2}$. The 30 images with $b > 0$ all have equal non-zero $\|\mathbf{G}\|$, Δ and δ, but unique orientation of $\mathbf{G}$. Thus we vary only 2 of the 5 degrees of freedom in the measurement pulse sequence. The set of gradient directions is chosen so that they are evenly distributed over the surface of the sphere. Each image is sensitive to diffusion in a particular direction, so overall we sample all directions as completely as we can without favouring or neglecting any particular orientations. If you look carefully, you can see differences in the $b > 0$ images that reflect the difference in sensitivity. This is a typical acquisition protocol, i.e. set of $\mathbf{y}_i, i = 1, \cdots, K$, for diffusion tensor imaging (DTI), which we shall come to later.

It is good practice always to produce a picture of a data set like this when you first come across it to check that its appearance corresponds with your intuition: do your understanding of the data and the basic models you plan to use make sense? If not, back to square one: either the data is corrupt in some way or, more likely, your model is missing an important effect.

3.4 Model fitting

Constructing a map of estimates of the parameter d is straightforward when we have just two images, but how do we do it when we have 33? One option is simply to average all the $b = 0$ images, separately average all the $b > 0$ images, then use Eq. 49, as we did before. This actually works fine, but ignores the fact that we have more measurements at $b > 0$ than at $b = 0$. Alternatively, we can write the problem as a matrix equation and solve as a linear system of equations in two unknowns. Again, taking logs of Eq. 48, our model says

$$\log S(b) = \log S(0) - bd. \tag{50}$$

We can treat the log measurements as our data, $\mathbf{x} = (\log S(0), d)$ as our unknowns, and solve the equation $\mathbf{A} = G\mathbf{x}$ where $\mathbf{A} = (\log A(b_1), \cdots, \log A(b_K))^T$ and the k-th row of the design matrix G is $(1, -b_k)$.

Does it make sense to treat $\log S(0)$ as an unknown? Yes! Despite the fact that we measure it directly, we only get noisy measurements; we estimate the signal. Figure 20 shows parametric maps of $S(0)$ and d obtained in this way. $S(0)$ is largely a nuisance parameter that we need to estimate d.

As well as eyeballing the raw data, to check our understanding of it, before starting to fit models, it is important to eyeball the fit of our model to the data before we start to make inferences and draw conclusions. Our model predicts the measurements from the estimates of $\log S(0)$ and d by plugging them back into Eq. 48, which we achieve all at once via the matrix equation $\mathbf{A} = G\mathbf{x}$. Figure 21 compares the model predictions with the measurements in three separate voxels, each representative of the different tissue types. The residual errors in the figure are $\sum(A - S)^2$ rather than $\sum(\log A - \log S)^2$. Since the model predicts that the diffusivity is the same in all directions, it predicts that the signal is the same for all the $b > 0$ measurements. We see that the data in fact shows considerable variation among those measurements, but is it noise or a genuine effect in the signal? We get some clues from looking at the residual fitting errors. In both CSF and grey matter voxels, where we expect the diffusion to be approximately isotropic, we get similar residual errors. In white matter, where we expect anisotropic diffusion, we observe much higher residual error suggesting that the model fits the data less well.

3.5 Diffusion tensor imaging

Let's move on to a more interesting model. Diffusion tensor imaging is the standard diffusion MRI technique. It generalises the simple model we used above by allowing the diffusivity to vary as a function of direction. Specifically, it models the diffusivity as a quadratic function of direction:

$$d(\hat{\mathbf{q}}) = \hat{\mathbf{q}}^T D \hat{\mathbf{q}}, \tag{51}$$

where

$$D = \begin{pmatrix} D_{xx} & D_{yx} & D_{xz} \\ D_{xy} & D_{yy} & D_{yz} \\ D_{xz} & D_{yz} & D_{zz} \end{pmatrix}. \quad (52)$$

is the diffusion tensor, which is a positive-definite symmetric 3×3 matrix. The model for the signal now depends on both the b-value and the gradient orientation, $\hat{\mathbf{q}}$:

$$S(b, \hat{q}) = S(0,0) \exp(-b\hat{\mathbf{q}}^T D \hat{\mathbf{q}}). \quad (53)$$

The diffusion tensor model describes the dispersion of water molecules as a zero-mean trivariate Gaussian distribution on the displacement of water molecules. The dispersion pattern thus has ellipsoidal contours. The eigenvalues of D determine the size and shape of the ellipsoidal contours and the eigenvectors determine its orientation. Different tissue types have characteristic diffusion tensor size and shape. In CSF, we expect large isotropic diffusion tensors (all eigenvalues large and approximately equal); in grey matter small isotropic diffusion tensors. In white matter, we expect anisotropy. Normally prolate anisotropic diffusion tensors (one large and two small eigenvalues giving cigar-shaped contours) when all fibres are aligned, but we may also see oblate diffusion tensors (two large and one small eigenvalues leading to pancake shaped contours), for example where orthogonal fibre populations cross.

If we take logs of both sides of the model in Eq. 53, we obtain an equation that relates the unknown parameters, $\log S(0,0)$ and the elements of the diffusion tensor, linearly to the log measurements:

$$\log S(b, \hat{\mathbf{q}}) = \log S(0,0) - b(q_x^2 D_{xx} + 2q_x q_y D_{xy} + 2q_x q_z D_{xz} + q_y^2 D_{yy} + 2q_y q_z D_{yz} + q_z^2 D_{zz}), \quad (54)$$

where $\hat{\mathbf{q}} = (q_x, q_y, q_z)^T$. So we can solve the matrix equation $\mathbf{A} = G\mathbf{x}$, where $\mathbf{A}$ is the vector of log measurements,

$$\mathbf{x} = (\log S(0,0), D_{xx}, D_{xy}, D_{xz}, D_{yy}, D_{yz}, D_{zz})^T \tag{55}$$

is our vector of unknown parameters, and the design matrix G has k-th row

$$(1, -b_k q_{kx}^2, -2b_k q_{kx} q_{ky}, -2b_k q_{kx} q_{kz}, -b_k q_{ky}^2, -2b_k q_{ky} q_{kz}, -q_{kz}^2), \tag{56}$$

b_k and $\hat{\mathbf{q}}_k$ are the device settings for the k-th measurement. Thus we solve for the diffusion tensor elements and $\log S(0,0)$ in every voxel of the image via Eq. 27.

The elements of the diffusion tensor themselves are not particularly informative to form a parametric map from. More commonly, we try to derive rotationally invariant features, such as the mean diffusivity

$$MD = (D_{xx} + D_{yy} + D_{zz})/3 \tag{57}$$

and the fractional anisotropy

$$FA = \frac{3}{2}\sqrt{\frac{\sum(\lambda_i - MD)^2}{\sum \lambda_i}}, \tag{58}$$

where $\lambda_i, i = 1, 2, 3$ are the eigenvalues of the diffusion tensor. We can also map fibre orientation by visualising the primary eigenvector orientation. Figure 22 shows examples of such parameter maps.

Figure 23 compares the model predictions with the measurements in a similar way to figure 21 and shows that we get a substantially better prediction of the data from the diffusion tensor model than the simple model we used previously even in grey matter and CSF where the anisotropy is low.

3.6 Noise model

The simple linear fit above assumes independent identically distributed Gaussian noise on each data point or element of $\mathbf{A}$, which are the log measurements. A reasonable first approximation (we shall improve on this shortly) is to assume that the variance of the noise on each measurement is the same. What does this mean for the log measurements we actually fit to?

We can write each measurement

$$A = S + \delta S \tag{59}$$

where δS is a normally distributed random variable with standard deviation σ. Thus

$$\log A = \log(S + \delta S). \tag{60}$$

Consider the Taylor expansion of the right hand side of the equation above:

$$\log(S + \delta S) = \log S + \delta S \frac{\partial}{\partial S} \log S + O(S^2) \tag{61}$$

$$\approx \log S + \frac{\delta S}{S} \tag{62}$$

The standard deviation of $\delta S/S$ is σ/S, which suggests that the standard deviation of each log measurement depends inversely on the underlying signal, S. Since the signals are not all equal, neither are the noise standard deviations. For this reason, most practitioners of DTI use weighted linear least squares to fit the diffusion tensor with the diagonal elements of the weighting matrix W in Eq. 39 set so that $W_{ii} = A_i^2$. That gives more stable and less biased estimates. **NOTE 16: Measurement is the best available guess of the underlying signal, which is what we should weight by; you can also iterate this procedure replacing the measurement**

with the model prediction from the previous iteration. Check if this algorithm has a name or appears formally anywhere.

NOTE 17: Error propagation perhaps elsewhere.

~~The method we used above to calculate the standard deviation of $\log A$ is~~ a simple example of a more general method called error propagation. One application of error propagation is to estimate the standard deviation of some function of a Gaussian distributed random variable; we can replace the log function in Eq. 61 with any general function f and see that the standard deviation of $f(A)$ is

$$\sigma \frac{\partial f}{\partial A} \tag{63}$$

where σ is the standard deviation of A.

The weighted linear least squares fit still makes an assumption of Gaussian noise, albeit with different variance on each log measurement. Gaussian noise on the measurements themselves is a much better model, but to estimate D under a Gaussian noise model, we must minimise $\sum(S_i - A_i)^2$ rather than $\sum(\log S_i - \log A_i)^2$, as we do above. Some DTI practitioners use non-linear optimisation to minimise $\sum(S_i - A_i)^2$, because it potentially reduces bias from assuming the wrong noise model; many of the standard DTI software packages implement non-linear diffusion tensor fitting. However, the disadvantage is that the approach adds all the baggage associated with non-linear optimisation: local minima, long run times, etc.

In fact, even minimising $\sum(S_i - A_i)^2$ is an approximation. A more precise model is the Rician distribution (or other distributions such as the non-central chi-squared distribution for parallel imaging **NOTE 18: check**).

The MR signal is actually complex valued and the noise on both the real and imaginary parts is Gaussian. The image intensity is actually the modulus of this complex measurement. The Rician distribution is the distribution we get from the modulus of a complex number with Gaussian distributed real and imaginary parts. We can see the distribution is not the same as the Gaussian, because it can never go negative. For large signal, in fact it is almost indistinguishable from a Gaussian, but it becomes different for low signals (within a few standard deviations of zero). Using non-linear optimisation and maximum likelihood estimation, we can estimate the parameters instead using a Rician noise model.

3.7 Parameter constraints

One advantage of non-linear fitting of linear methods is that it allows us to impose constraints on the diffusion tensor. A key constraint on the diffusion tensor is that it is positive definite. Linear estimation often results in unrealistic diffusion tensors that have negative eigenvalues. An elegant and efficient way to impose the constraint is to use the Cholesky decomposition. It exploits the fact that any positive definite symmetric matrix can be written as LL^T where L is a lower triangular matrix, i.e. has the form

$$L = \begin{pmatrix} L_{xx} & 0 & 0 \\ L_{xy} & L_{yy} & 0 \\ L_{xz} & L_{yz} & L_{zz} \end{pmatrix}. \tag{64}$$

Thus we use the transformation method and optimise the elements of L and subsequently reconstruct the diffusion tensor as $D = LL^T$.

3.8 Summary

The two models I have discussed here are only the most basic used in diffusion MRI. A wide variety of alternatives exist, see (Panagiotaki et al NeuroImage 2012) for example and many provide access to more interesting features of the tissue such as cell size and packing density. We will return to this in later sections and see some examples. The coursework associated with this document describes one in particular, the ball and stick model, which provides parameters with more direct cellular interpretation that express the density and orientation of axons in white matter.

Figure 15: Example of parametric mapping. The overlay shows a map of axon density in the corpus callosum of a fixed monkey brain. The axon density is not observed directly, but fitted to a collection of image intensities in each voxel of the image.

Figure 16: Tissue types in the brain.

Figure 17: The pulsed-gradient spin-echo pulse sequence used for diffusion MRI.

Figure 18: Apparent diffusivity map estimated from two images, $b = 0$ and $1000\,\mathrm{smm}^{-2}$.

Figure 19: A slice through each image comprising an example diffusion MRI data set.

Figure 20: Apparent diffusivity map from the full data set.

Figure 21: Comparison of raw measurements with those predicted by the simple ADC model in three example voxels.

Figure 22: Maps of mean diffusivity, fractional anisotropy, and DT principle direction from diffusion tensor imaging.

Figure 23: Comparison of raw measurements with those predicted by the diffusion tensor model in the same three example voxels as used in Fig. 21.

4 Uncertainty

In the previous section, we have seen how maximum likelihood techniques provide the estimate that maximises $p(\mathbf{A}|\mathbf{x})$ and Bayesian techniques the estimate that maximises $p(\mathbf{x}|\mathbf{A})$. However, just because $\tilde{\mathbf{x}}$ is the most likely solution (in some sense), does not make it right and nearby solutions $\tilde{\mathbf{x}} + \delta\mathbf{x}$ wrong; other solutions are just less likely given the available information. Often it is important to quantify how much less likely other candidate solutions are, so that we can express our confidence in the single most likely parameter estimate. Estimating that confidence is the topic of this section.

We can express the confidence in parameter estimates in various ways. The most complete expression is the full posterior distribution and we focus in this section on reconstructing that distribution. Thus, we assume that the objective function for parameter estimation is $p(\mathbf{x}|\mathbf{A})$, i.e. we adopt the Bayesian approach. However, all the techniques we discuss work equally well for reconstructing $p(\mathbf{A}|\mathbf{x})$, i.e. within the maximum likelihood framework. After all, the two are equivalent (at least in practice if not strictly in theory) if we choose the prior $p(\mathbf{x})$ as a broad uniform distribution covering the full range of possible settings of each parameter. From a reconstruction of $p(\mathbf{x}|\mathbf{A})$, we can derive various simpler statistics, such as the covariance of $\mathbf{x}$ or simply the standard deviation of each x_i, or confidence intervals stating for example the range of values within which x_i lies with probability of, say, 95%.

We start with a couple of applications in biomedical imaging to illustrate the importance of estimating confidence in parameter estimates. Then we will cover three broad classes of technique for estimating $p(\mathbf{x}|\mathbf{A})$ and simple de-

rived statistics of confidence: Laplace's method, bootstrapping, and Markov Chain Monte Carlo (MCMC).

4.1 Parametric mapping

A lot of work on parametric mapping never considers confidences, but works solely with the MLE or MAP parameter estimates. Figure 24 shows parametric maps from a technique called quantitative magnetisation transfer (qMT). It is another MRI technique, which this time sensitises the image intensity to the exchange of protons between macromolecules, such as the fatty myelin sheaths of axons in white matter, and free pools of water, such as in the protoplasm inside cells. The signal change is important because it reflects the density of the macromolecules. Myelin is a fatty substance surrounding axons in nervous tissue, such as white matter in the brain. Myelin insulates the axons allowing them to conduct electrical signals more efficiently increasing the transmission speed of messages. Diseases like multiple sclerosis cause damage to the myelin sheaths around axons disrupting the communication channels in the brain. By fitting appropriate models, we can estimate the fraction of protons attached to myelin and infer myelin density, as well as several other useful parameters. Figure 24 comes from a standard parametric mapping procedure: various MR images are acquired that support fitting a simple model of magnetisation transfer to estimate the parameters in every voxel via maximum likelihood estimation. The authors then examine the mean and variance of each parameter of various distinct anatomical regions in the brain. This is fairly typical of early stage development of a new parametric mapping technique. Later stages might then compare the statistics of maximum likelihood estimates in various regions between different groups of

subjects, such as a patient population and an age-matched set of controls. It is unusual for work at any of these stages to consider confidence intervals of their parameter estimates. Why not and does it matter? One might argue that averaging parameter values over extended regions of interest mitigates the need. This makes the implicit assumption that natural variation in the actual parameter value from voxel to voxel dominates the estimation error. Perhaps a violation of the assumption is likely to be clear in parameter maps, which would appear noisy. However, despite common practices, it is wise always to check the validity of these hidden assumptions by evaluating some confidence statistics.

4.2 Tractography

NOTE 19: Separate as example at end of chapter?

Modelling the estimation error becomes more important when we need to consider individual voxels in isolation or small regions of a few voxels. This can become particularly important for post-processing algorithms that use parametric maps to make downstream inferences. An application of diffusion imaging called tractography illustrates this nicely.

Diffusion imaging has two main applications. One is parametric mapping, as we saw in the previous section. The other is for mapping anatomical brain connectivity using a process called tractography. Figure 25 illustrates the kind of output we can get from diffusion MRI-based tractography. Tractography is the process of reconstructing trajectories of white matter pathways from diffusion MRI data. The simplest algorithms start from a parametric map of the dominant fibre orientation. That might come from the principle

direction of the diffusion tensor, or the orientation parameter of the ball-and-stick model **NOTE 20: see coursework**, or a corresponding parameter of many other models; broadly the direction of greatest water mobility gives us an estimate of the local fibre orientation. Algorithm 1 gives an outline of the basic procedure, which is known as *deterministic tractography*.

Pick a starting location $\mathbf{r}_0$.

$i = 0$

while $\mathbf{r}_i \in \Omega$ **do**

  Get the local fibre orientation $\mathbf{n}_i$

  Set $\mathbf{r}_{i+1} = \mathbf{r}_i + \delta \mathbf{n}_i$, where δ is a small step size.

  Set $i = i + 1$

end

Return the list $\{\mathbf{r}_0, \cdots, \mathbf{r}_i\}$ of points on the connection trajectory.

Algorithm 1: Deterministic tractography. The set Ω is the region of the image in which the fibre orientation estimate exists; this might be the region corresponding to the brain, or may be further limited to white matter areas.

Figure 26 illustrates the procedure.

Simply following fibre orientations from point-to-point through the image allows us to reconstruct a global trajectory and infer connectivity of the end points via the reconstructed path. This simple idea has had a major impact in neuroscience over the last decade or so and provided a whole new level of understanding of brain anatomy and function in health and disease. The only way previously we could study the wiring of the brain was through painstaking dissection of post-mortem brains. Diffusion imaging and tractography provide a non-invasive probe into whole brain connectivity of live subjects from just 10-30 minutes lying in an MRI scanner.

One problem with the simple algorithm above is that the fibre orientation

estimates are uncertain. Figure 27 shows a map illustrating the uncertainty in the maximum likelihood fibre orientation estimate over a slice of the brain and shows that the uncertainty is high in isotropic regions, where the orientation is poorly defined. It is lower in white matter regions, but still significant, particularly in areas where fibres cross. This is important, because small fluctuations in one step along the pathway a tractography algorithm follows can have a large downstream effect. Figure 28 shows an illustrative example of how this can happen. Knowledge of the uncertainty in our individual fibre orientation estimates enables us to probe the full set of possible pathways and evaluate their relative likelihood. This is the basis of *probabilistic tractography*, which repeatedly resamples the fibre orientation estimates from the posterior distribution on the orientation at each voxel and repeats the procedure in Algorithm 1; figure 29 illustrates the procedure. Accounting for the uncertainty in this way can provide a much more complete picture of brain connectivity than Algorithm 1 alone. The Behrens and Jbabdi chapter of the Johansen-Berg and Behrens Diffusion MRI book give a more detailed account of probabilistic tractography.

4.3 Laplace's method

Laplace's method is a general procedure for approximating intractable integrals that involves approximating the integrand with a Gaussian function. In a similar way in parameter estimation, the approximation of $p(\mathbf{x}|\mathbf{A})$ as a Gaussian is sometimes referred to as Laplace's method. The mean of the Gaussian distribution is the maximum likelihood or MAP estimate, $\tilde{\mathbf{x}}$. The covariance comes directly from the Hessian of the log posterior at the opti-

mum:

$$\Sigma = -\left(\frac{d^2}{d\mathbf{x}^2}\log p(\tilde{\mathbf{x}}|\mathbf{A})\right)^{-1} = -(H(\tilde{\mathbf{x}}))^{-1}. \quad (65)$$

This is straightforward to see mathematically, since if $p(\mathbf{x}|\mathbf{A})$ is Gaussian,

$$\log p(\mathbf{x}|\mathbf{A}) = -\frac{1}{2}(\mathbf{x}-\tilde{\mathbf{x}})^T\Sigma^{-1}(\mathbf{x}-\tilde{\mathbf{x}}) + \text{const.} \quad (66)$$

so

$$\frac{\partial^2}{\partial x_i \partial x_j}\log p(\mathbf{x}|\mathbf{A}) = -(\Sigma^{-1})_{ij} \quad (67)$$

The intuition is that $H(\tilde{\mathbf{x}})$ quantifies the curvature of the objective function at the optimum. If H is low, the curvature is low, so we have a broad flat peak around $\tilde{\mathbf{x}}$. Thus a small perturbation $\tilde{\mathbf{x}} + \delta\mathbf{x}$ has similar likelihood to $\tilde{\mathbf{x}}$, and the variance of $p(\mathbf{x}|\mathbf{A})$ is high, so we have low confidence in our MAP estimate. On the other hand, if H is high, the function is sharply peaked at the optimum, suggesting that $\tilde{\mathbf{x}} + \delta\mathbf{x}$ has significantly lower likelihood than $\tilde{\mathbf{x}}$ and that the variance of $p(\mathbf{x}|\mathbf{A})$ is low - we have high confidence in the MAP estimate.

Although the Hessian matrix can be fiddly to compute, as discussed in section 2.2, many gradient descent algorithms compute it during their execution, so we often get it for free. For precisely the reason that it provides a useful estimate of the uncertainty in our parameter estimates, most implementations of these algorithms, including `fminunc` in matlab, will return the value of the Hessian matrix at the optimum point they converge on.

We can derive a variety of useful statistics about confidence in our parameter estimates from the Gaussian model of $p(\mathbf{x}|\mathbf{A})$. For example, if we consider just one parameter x_i, the i-th diagonal element of Σ^{-1}, $\sigma(x_i) = (\Sigma^{-1})_{ii}$, is the standard deviation of the posterior on that parameter. Often we might quote the "two-sigma range", which is $[\tilde{\mathbf{x}} - 2\sigma(x_i), \tilde{\mathbf{x}} + 2\sigma(x_i)]$. That range contains

the true $\mathbf{x}$ with about 95% probability; the one-sigma range corresponds to about 68% and the three-sigma range to about 99%.

Note of caution: the standard SSD objective function,

$$f = \sum_{i=1}^{K} (A_i - S_i)^2, \tag{68}$$

drops the scaling factor $(-2\sigma^2)^{-1}$ in $\log p(\mathbf{A}|\mathbf{x})$. We need put that factor back by dividing the Hessian returned by, say, `fminunc` by $(2\sigma^2)^{-1}$ to make sure we have the right scale for the standard deviation of the parameter posteriors (the minus sign disappears, because f is proportional to $-\log p(\mathbf{A}|\mathbf{x})$).

4.4 Bootstrapping

Boostrapping is a resampling technique that aims to use the data directly to estimate its own distribution and that of the model parameters. The name comes from the phrase "pull yourself up by your bootstraps", which expresses the notion of getting something for nothing, to indicate the apparent circularity of the concept. However, the endeavour is less of a waste of time than the name might suggest and it is a powerful and widely used statistical technique.

Bootstrapping algorithms all follow the same basic procedure. The t-th iteration, $t = 1, \cdots, T$, constructs a bootstrap data set, $\hat{\mathbf{A}}_t$, from the original set of measurements, $\mathbf{A}$, and fits the model to that data set to obtain a parameter estimate $\tilde{\mathbf{x}}_t$. The output is the list $\{\tilde{\mathbf{x}}_1, \cdots, \tilde{\mathbf{x}}_T\}$ of samples of $p(\mathbf{x}|\mathbf{A})$.

From the list of samples of $\tilde{\mathbf{x}}$ we can construct any statistic of $p(\mathbf{x}|\mathbf{A})$ in a similar way to the Gaussian approximation we get from Laplace's method.

We can fit a Gaussian to the samples to estimate the covariance and thus produce two-sigma ranges in much the same way. However, the bootstrap samples give us a more general picture of the shape of $p(\mathbf{x}|\mathbf{A})$ without the Gaussian assumption. We can visualise $p(\mathbf{x}|\mathbf{A})$ directly by plotting a histogram of the samples. If the shape differs significantly from Gaussian, the two-sigma range is not a good description of the range of sensible estimates for $\mathbf{x}$. Suppose for example, the distribution is skewed in one direction, then the range should not be symmetric about the mean. We can still get a sensible 95%-range from the bootstrap samples of a particular parameter by ranking all the estimates and taking the sample at position $0.025T$ in the list as one end of the range and that at position $0.975T$ as the other. Figure 30 illustrates these ideas for a particular parameter.

The variants of the bootstrap procedure differ in the way the bootstrap samples are created. The following subsections go through a range of common variants and outline their assumptions, pros and cons. Statistical texts such as Davison and Hinkley provide much more detailed information, but we focus here on the basics of using these ideas and interpreting their output.

4.4.1 Parametric bootstrap

Parametric bootstrap is conceptually the most straightforward procedure. We start by fitting the model to the original data set $\mathbf{A} = (A_1, \cdots, A_K)$. From the best fit parameters, we predict noise free signals $\mathbf{S} = (S_1, \cdots, S_K)$. For each bootstrap sample, we then use a noise model to sample noise perturbations $\mathbf{E} = (\eta_1, \cdots \eta_K)$, and obtain the bootstrap sample by adding the

perturbations to the signals:

$$\hat{\mathbf{A}}_t = \mathbf{S} + \mathbf{E}_t. \tag{69}$$

The key assumption implicit in the parametric bootstrap is that the noise model is correct. If it is not, then the perturbations of the signal are unrealistic and the samples are not representative of the true posterior. We can potentially estimate the parameters of the noise model from the residual fitting errors, provided we have many more measurements than model parameters so we can be sure we are not overfitting the data making the residuals unrealistically low. For example, if we assume independent identically distributed Gaussian noise, we can estimate the variance as

$$\sigma^2 = \frac{1}{K-N} \sum_{k=1}^{K} R_k^2, \tag{70}$$

where R_i is the residual error of the i-th measurement. Note that for an unbiased estimate of the variance, we need to normalise the sum of square differences by $K - N$ where N is the number of fitted model parameters. In fact a better estimate comes from

$$\sigma^2 = \frac{1}{K-1} \sum_{k=1}^{K} \frac{R_k^2}{1 - H_{kk}}, \tag{71}$$

where here $H = G(G^T G)^{-1} G^T$ is the so-called hat matrix. **NOTE 21: See the Davison and Hinkley bootstrapping book to get an understanding of this.**

All the remaining bootstrap variants are examples of the non-parametric bootstrap, which resample the data directly.

4.4.2 Classical bootstrap

The classical bootstrap constructs a bootstrap sample by drawing K samples from $\mathbf{A}$ at random with replacement. Thus

$$\hat{A}_{tk} = A_{\lfloor U(1,K+1) \rfloor}, \tag{72}$$

where $k = 1, \cdots, K$ and $U(1, K+1)$ is a uniformly distributed random variable on the interval $(1, K+1)$ so that the index $\lfloor U(1, K+1) \rfloor$ is a uniformly distributed integer in the range $1, \cdots, K$.

The advantage of the classical bootstrap over the parametric bootstrap is that it does not require assumptions about the noise model, but uses the variation of the data to capture it.

A limitation of the classical bootstrap is that it disrupts the design of the original experiment, which may cause it to overestimate the uncertainty of the parameter estimate from the original data set. Suppose we are fitting a line to well spaced data points. One draw might end up with only repeats for neighbouring samples. The error on fitting a line to neighbouring points is higher than to remote points, because the experiment design is less good. Figure 31 illustrates the problem.

4.4.3 Repetition bootstrap

The repetition bootstrap requires several repeat measurements of each data point. Suppose we acquire R repeats of each of our K measurements, so the

full data set available for bootstrapping is

$$\begin{array}{cccc} A_{11}, & A_{12}, & \cdots, & A_{1R} \\ A_{21}, & A_{22}, & \cdots, & A_{2R} \\ \vdots & \vdots & & \vdots \\ A_{K1}, & A_{K2}, & \cdots, & A_{KR} \end{array} \tag{73}$$

Each bootstrap sample contains one measurement from each group of R:

$$\hat{A}_{tk} = A_{k\lfloor U(1,R+1)\rfloor}. \tag{74}$$

The key advantages of the repetition bootstrap is that it retains the design of the intended experiment, unlike the classical bootstrap, while still sampling the noise distribution from the data, like the classical bootstrap. The downside is that it is expensive in terms of data acquisition, as it requires R times the number of measurements as the actual experiment requires.

4.4.4 Residual bootstrap

The residual bootstrap resamples the residuals rather than the data points. We start by fitting the model to the data, as in the parametric bootstrap, predict the signal $S(\tilde{\mathbf{x}}; \mathbf{y}_k), k = 1, \cdots, K$ corresponding to each data point from the fitted model, and compute the residuals for each data point, $R_k, k = 1, \cdots, K$. Then, to construct a bootstrap sample, for each data point we draw a residual at random, with replacement, from the set $\{R_1, \cdots, R_K\}$ and add it to the predicted signal. Thus

$$\hat{A}_{tk} = S(\tilde{\mathbf{x}}; \mathbf{y}_k) + R_{\lfloor U(1,K+1)\rfloor}. \tag{75}$$

The advantage of the residual bootstrap is that it retains the intended experiment design without requiring additional acquisitions. However, the limita-

tion is that it assumes that the noise statistics are the same on all data points. It also requires that the number of measurements is much greater than the number of model parameters to avoid overfitting, which would give unrealistically low residuals. For example, if we fit a model with three parameters to three data points, the residual error is always zero, so the residual bootstrap process will show no variation although of course there is uncertainty in the parameter estimates.

4.4.5 Wild bootstrap

The wild bootstrap multiplies each residual by a standard normal distribution:

$$\hat{A}_i = S(\tilde{\mathbf{x}}; \mathbf{y}_i) + R_j N(0, 1). \tag{76}$$

The advantage of the wild bootstrap over the residual bootstrap is that it accommodates different noise models on each data point. However, it underestimates noise on points that just happen to be close to the fitted value. It also imposes some statistical properties on the noise distribution. So long as the number of measurements is sufficiently large compared to the number of model parameters, the noise properties fall out and the wild bootstrap can perform well in practice.

4.4.6 Practical issues

The particular choice of bootstrap strategy really depends on the application; different choices are appropriate for different situations and we have to weigh up the pros and cons listed above depending on the data, the estimation problem, and precisely what uncertainty we want to evaluate and why.

A common observation in results from the repetition bootstrap is that it underestimates the variance in parameter estimates, particularly when the number of repeats is small. This is expected because it draws samples from a limited superset rather than the full general population of possible measurements. Various adaptations, such as the "bootknife" procedure [**?**] aim to avoid that downward bias in the variance. In the bootknife algorithm, at each iteration of the bootstrap procedure, one measurement is removed from the data pool. **NOTE 22: Not clear exactly how this works from applications like Chung Neuroimage 2006 and Jeurissen thesis chapter 5...**

Another issue to consider in applications of bootstrap is how many samples are required. Again, this depends on the application and precisely what statistics of the posterior distribution we are trying to estimate: we need fewer samples to get a good estimate of a parameter like the standard deviation of the posterior than a more subtle statistic like its fourth moment. Figure 32 shows the convergence of various statistics of the bootstrap list as a function of iterations. It is always worthwhile to plot pictures like this to have confidence that the choice of T, the number of iterations, is reasonable.

4.5 Markov Chain Monte Carlo

MCMC is a beautifully simple algorithm for sampling a density function when its form is unknown and, more specifically, we do not know its normalising constant, but we can sample its unnormalised value at any point. This is precisely the situation in parameter estimation where we would like to reconstruct the density function $p(\mathbf{x}|\mathbf{A})$. We can evaluate the likelihood,

$p(\mathbf{A}|\mathbf{x})$, and the prior $p(\mathbf{x})$ in Eq. 21, but we do not know the normalising constant

$$p(\mathbf{A}) = \int p(\mathbf{A}|\mathbf{x}) d\mathbf{x}. \tag{77}$$

MCMC enables us to sample and reconstruct $p(\mathbf{x}|\mathbf{A})$ without having to evaluate $p(\mathbf{A})$.

The algorithm produces a list of samples in a similar way to bootstrapping, although the procedure is quite different. Algorithm 2 outlines a basic procedure, although many variations and refinements are in use in practice; see for example the compilation by Gilks et al (Chapman-Hall 1996).

$\mathbf{x}_0$ = some starting point.;
for $t = 1, ..., T$, **do**
    $\mathbf{x}_c = Q(\mathbf{x}_{t-1})$;
    if $\alpha(\mathbf{x}_c, \mathbf{x}_{t-1}) > U(0,1)$ **then**
        $\mathbf{x}_t = \mathbf{x}_c$;
    else
        $\mathbf{x}_t = \mathbf{x}_{t-1}$;
    end
end
Return the list $\{\mathbf{x}_1, \cdots, \mathbf{x}_T\}$ of samples of $p(\mathbf{x}|\mathbf{A})$.;
;

Algorithm 2: The Metropolis-Hastings algorithm for Markov chain Monte Carlo. $U(0,1)$ is a random draw from the uniform distribution on $(0,1)$. $Q(\mathbf{x})$ is a random draw from a proposal distribution $q(\cdot|\mathbf{x})$; the ratio α is defined in the text.

The algorithm maintains a current position in the domain on which the density function we are interested in is defined; here a current set of parameter

estimates $\mathbf{x}_i$. At each step, the algorithm selects a candidate new position $\mathbf{x}_c$ randomly from a *proposal distribution* $q(\cdot, \mathbf{x}_i)$. The proposal distribution can take almost any form, but often it is such that $\mathbf{x}_c$ is a perturbation of the current position, for example, $q(\cdot, \mathbf{x}) = N(\mathbf{x}, \sigma)$. The next step decides whether to accept the new position as $\mathbf{x}_{t+1}$ or to stay at the previous position so that $\mathbf{x}_{t+1} = \mathbf{x}_t$. The algorithm accepts the $\mathbf{x}_c$ with probability $\min(\alpha(\mathbf{x}_c, \mathbf{x}_t), 1)$, where

$$\alpha(\mathbf{x}, \mathbf{y}) = \frac{p(\mathbf{x}|\mathbf{A})q(\mathbf{x}|\mathbf{y})}{p(\mathbf{y}|\mathbf{A})q(\mathbf{y}|\mathbf{x})} = \frac{p(\mathbf{A}|\mathbf{x})p(\mathbf{x})q(\mathbf{x}|\mathbf{y})}{p(\mathbf{A}|\mathbf{y})p(\mathbf{y})q(\mathbf{y}|\mathbf{x})} \tag{78}$$

For a symmetric proposal distribution, i.e. $q(\mathbf{x}|\mathbf{y}) = q(\mathbf{y}|\mathbf{x})$, such as a Gaussian on the real line, α reduces to the ratio of the posteriors of the two candidate points $\mathbf{x}_c$ and $\mathbf{x}_t$. (Note that the intractable denominator in Eq. 21 cancels, as the right-hand side of Eq. 78 demonstrates.) In that case, if the new position has higher posterior likelihood than the old, so that $\alpha > 1$, we always accept the new position. However, if the new position has lower likelihood, then $\alpha < 1$ and we accept it anyway with probability α: if the new position is much less likely, i.e. $\alpha \approx 0$, we are unlikely to accept it, but if it has similar likelihood, $\alpha \approx 1$, and we are very likely to accept $\mathbf{x}_c$. For asymmetric proposal distributions, we must weight the ratio by the likelihoods of making the steps in each direction, as shown in the general expression in Eq. 78. This can arise, for example, if we are sampling a distribution on a non-Cartesian space. Suppose, that two of the parameters of interest are spherical polar co-ordinates θ and ϕ encoding a position on the sphere, so that $\mathbf{x} = (\theta, \phi)$. If we set the proposal distribution to be normally distributed with mean zero for each parameter, the proposal is not symmetric, because there are many more points on the sphere with $\theta = \pi/2$, i.e. at the equator of the sphere, than $\theta = 0$, i.e. at the pole. The relative likelihood of any particular θ is $\sin(\theta)$, which we include as factors in the numerator and

denominator to rebalance α.

The stochastic element to the algorithm allows it to make downhill moves and thus, for example, move between different modes of a multimodal distribution. Over a large number of iterations, the amount of time the algorithm spends in each region of the domain is proportional to the likelihood of the parameters at that point, so the procedure samples the density function.

The algorithm has various settings to tune:

- The *burn-in*. MCMC is guaranteed to converge to sampling the target distribution, here $p(\mathbf{x}|\mathbf{A})$, but it may take some time to get there. In particular, the process can start from any arbitrary starting point, which means that the early samples can be far away from the main support of the target distribution. For that reason, we normally discard some number, B, of samples from the beginning of the list, which is known as the "burn-in" period.

- The *sampling interval*. Consecutive $\mathbf{x}_i$ in the list returned by the algorithm are usually dependent, because the step size between consecutive points is generally smaller than the main support of the density function. Sometimes this dependence is unimportant, because collectively the full set of samples still provide a good picture of the target distribution so long as the algorithm is run for a sufficiently large number of iterations. However, some applications, such as probabilistic tractography discussed above, require independent samples. We can achieve this by discarding a number of intermediate steps between each sample returned. So, after the burn in, only every I-th sample in the complete list joins the final list of samples returned by the algorithm. We can

write the full list of samples:

$$
\begin{aligned}
&\mathbf{x}'_1 \\
&\vdots \\
&\mathbf{x}'_B \\
&\mathbf{x}'_{B+1} &&= \mathbf{x}_1 \\
&\mathbf{x}'_{B+2} \\
&\vdots \\
&\mathbf{x}'_{B+I} \\
&\mathbf{x}'_{B+I+1} &&= \mathbf{x}_2 \\
&\mathbf{x}'_{B+I+2} \\
&\vdots \\
&\mathbf{x}'_{B+2I} \\
&\mathbf{x}'_{B+2I+1} \\
&\vdots \\
&\mathbf{x}'_{B+TI} \\
&\mathbf{x}'_{B+TI+1} &&= \mathbf{x}_T.
\end{aligned}
$$

- The *number of samples*, T. The number of samples determines the precision of statistics derived from the chain. For example, we get a better estimation of the mean of $p(\mathbf{x}|\mathbf{A})$ from a chain of 100000 samples than we do from a chain of 1000 samples, but it takes 100 times longer to compute.

- The *proposal distribution*, $q(\mathbf{x})$. In theory, the form of q can be almost arbitrary and the MCMC process still samples the target distribution, although the choice of q affects convergence rates. In practice, simple

choices, such as a Gaussian distribution centred on the current sample, i.e. $q(\mathbf{x}) = N(\mathbf{x}, \Sigma)$, often work well. Various heavy-tailed distribution, such as the Laplace or t-distribution, can be advantageous for sampling multi-modal distributions, as they produce large jumps more often.

The key choice to make lies in the parameters of the proposal distribution. Suppose our proposal distribution is simply a product of Gaussians on each parameter $x_i, i = 1, \cdots, K$, i.e. $q(\mathbf{x}) = N(\mathbf{x}, \Sigma)$ with the covariance, Σ, diagonal. The variance of each dimension, i.e. each diagonal entry σ_{ii}^2 of Σ, must be chosen carefully to ensure good behaviour. If the variance is small compared to the value of x_i, steps are small so the value of $p(\mathbf{x}|\mathbf{A})$ changes little from current to candidate position. Thus the rate of acceptance is high, but it can take many steps to cover the whole range of the distribution causing long computation times. On the other hand, if we set the variance high, steps are large and often produce candidate positions with low likelihood, so the acceptance rates become low. Thus, again, the time taken to sample the whole distribution becomes large. In practice, we need to optimise this trade-off by finding intermediate variances producing step sizes that can cover the range of the distribution in a reasonable number of steps without exploring too much of the low-likelihood domain outside the main support of the distribution.

A good rule of thumb is to aim for $20 - 50\%$ acceptance rate. However, we must take care to tune the step size for each individual parameter, as the parameter for which the step size is largest (relative to the support of the posterior on that parameter) has most influence on the acceptance rate. For example, we can achieve a 30% acceptance rate by setting the step size very small for all but one parameter and tuning

the step size for that parameter to get 30%. This only provides good coverage of $p(\mathbf{x}|\mathbf{A})$ along one axis, because the small steps in the other parameters take a long time to traverse the full range. A simple recipe for tuning step sizes is to initialise all step sizes small so we get acceptance rate close to 100%. Then one-by-one tune the step size for each parameter to obtain $20 - 50\%$ acceptance with all others fixed small. The final runs use the tuned value for each parameter.

- The *number of chains*. Practitioners often run several MCMC "chains", which are separate runs with different starting points or random seeds. This can be helpful to determine when convergence has occurred, as the chains should then show similar statistics. One question that arises is then: how many chains do we need? Some argue that using one long chain is always preferable to multiple shorter ones, as it gives the process more opportunity to explore the full support of the target distribution.

In summary, despite its simplicity, the MCMC algorithm can take a fair amount of tuning to get working correctly and efficiently. To ensure good behaviour, always check the "chains", i.e. plot the trajectory of each x_i as a function of t, as that often reveals undesirable behaviour. Figure 33 illustrates this. It is always worth running some initial experiments which run MCMC for a very long time to get a complete picture of its behaviour in a particular application, before choosing final working parameter settings. Many strategies are available for evaluating useful diagnostics of MCMC chains, choosing proposal distributions, and generally for speeding up convergence. The various chapters of the Gilks et al book give very useful information on all these issues for those looking to use MCMC seriously.

4.6 Summary

In this chapter we have covered various methods for approximating the posterior distribution on unknown parameter values, which provides useful statistics of confidence in parameter estimates. Specifically, we have introduced Laplace's method, bootstrapping and MCMC. We cover just the basic methods and many more sophisticated techniques and variations of the techniques we cover are available in the wider literature. See for example, the text books on MCMC by Gilks et al and on bootstrapping by Davison and Hinkley. In imaging applications, these ideas serve several purposes, from estimating the voxelwise confidence in parameter maps to inform on how much variation we see arises from genuine variation in anatomy, to supporting statistical post-processing algorithms, such as tractography.

Figure 24: Parametric maps from quantitative magnetisation transfer imaging; from Cercignani et al.

Figure 25: Example of the output provided by a simple tractography algorithm.

Figure 26: Illustration of the deterministic tractography procedure.

Figure 27: Depicts cones of uncertainty on each fibre orientation estimate from diffusion tensor imaging in a region of the brain; from Jones et al 2003.

Figure 28: Ill-posedness of deterministic tractography at a branching fibre configuration.

Figure 29: Illustration of the probabilistic tractography procedure.

Figure 30: Shows the output from an example bootstrap execution.

Figure 31: Shows the disruption to the experiment design that can occur in the classical bootstrap.

Figure 32: Compares the convergence of two statistics of the posterior distribution as a function of number of bootstrap samples.

Figure 33: Example MCMC chains illustrating common problems.

5 Model selection

NOTE 23: Profile William of Ockham? Lived in the 14th century as a monk spending his time on logic and mathematics...

This section discusses the problem of choosing from a set of candidate models the one that best explains a set of measured data. We start with an illustrative example showing why the problem is important, then cover classical frequentist tests for deletion of variables (the F-test), a range of information criteria for model selection, Bayesian model selection, and cross validation.

5.1 Simulated example

As discussed in section 1, fluctuations in starlight intensity allow us to detect and estimate features of planets orbiting remote stars. However, once we detect fluctuations, how can we tell if they arise from just a single orbiting planet, two, several, or even are not the product of orbiting bodies at all, but just noise in our measurements. Figure 34 shows some light-intensity data measured from a star as a function of time. How many planets would you guess are orbiting this star?

We can construct a hierarchy of models to choose from with increasing numbers of planets. Model M_1 has one orbiting planet. To keep things simple, we shall assume its parameters are star luminance, orbit period and phase, distance from star, and planet size; five parameters in total. The red line shows the best fit of M_1 to the data, which achieves a sum of square differences score of 6.274×10^6. Model M_2 has two orbiting planet, so has additional parameters, which are the orbit period and phase, distance and size of the

second planet; nine parameters in total. The green line shows the best fit of M_2 to the data and the sum of square differences reduces to 5.508×10^6. It seems likely there are two planets orbiting this star, as we get a substantial reduction in fitting error from M_2 compared to M_1.

Systems with different numbers of planets require fundamentally different models, with different equations and different numbers of parameters. Thus, the number of planets is not simply a parameter of a more general model. The model selection problem is to identify the model from a set of candidates that best explains a particular data set. To do that, we need to fit each model and evaluate its likelihood given the data and potentially some prior knowledge. One plausible alternative strategy is to adopt a single model with more planets than we ever need. Suppose we allow for four planets and always use model M_4. When only one or two planets exist, we assume that the parameters of non-existent planets will reflect non-existence; for example, the size parameter ends up close to zero. We can then ignore parameters associated with components where the planetary size estimate is close to zero. This approach has several problems. First, we have no clear way to determine how close to zero the size has to be to declare the planet non-existent. Second, the model fitting becomes harder the more parameters we have to fit; local minima, parameter uncertainty, computation time, and numerical instability all increase as the number of parameters increases. Explicitly excluding irrelevant parameters usually helps stabilise fitting. Third, the overcomplexity of the model introduces redundancy: a single planet may manifest as one size estimate of, say, V and three of zero, but a solution with two equal components with size estimate $V/2$ and two zero sizes would fit the data equally well. That particular ambiguity is simple to identify and rectify, but other combinations may be less easy to disentangle. Sparse reconstruction

techniques potentially overcome these problems by including in the objective function that penalises the number of non-zero parameters. However, for now, the approach we adopt instead is to identify the most plausible model from a set of candidates.

Figure 35 shows a similar data set to figure 34 from a different star. As before the red and green lines show the fits of M_1 and M_2 to the data, respectively. The residual errors are 4.076×10^6 and 4.071×10^6, respectively; a very slight reduction in error score for M_2 compared to M_1. However, since M_1 is a special case of M_2, M_2 is guaranteed to fit the data at least as well as M_1 regardless of how many planets really exist. Model M_2 can always achieve the same residual error as M_1 by setting the size of one planet to zero and usually M_2 will do better by finding some spurious settings of the parameters of the second planet that produce a slightly improved error score simply by fitting the noise better. It seems likely in figure 35 that the improvement in M_2's residual error is a noise effect, since it is very small. Despite M_2 fitting the data better, most likely M_1 will predict future measurements better, because the random fluctuations that M_2 picks up on to produce a lower error score will be different in the future.

One last example in figure 36, where the error scores for M_1 and M_2 are 4.197×10^6 and 4.192×10^6, respectively. However, in both cases, the estimates for planet size are very small **NOTE 24: Need to include parameter estimates**. We can also fit a zero planet model, M_0, where the intensity is actually constant; the blue line shows the best fit of the zero-planet model, which has an error score of 4.442×10^6. For the two previous examples, it is fairly easy to tell by eye which of the candidate models is correct, but here we cannot be certain whether the differences in error score between M_0,

M_1 and M_2 are significant enough to select one model over the others. The tools introduced in this section provide principled ways of making this kind of decision and, potentially, combining evidence from multiple models.

5.2 Basic principles

A key idea in model selection is the law of parsimony or Occam's razor. William of Ockham generally gets attributed the philosophy that if you have a set of plausible explanations for something, the simplest one that works is probably correct, although apparently the idea predates him. However, this is a key guiding principle in model selection: the best model is the simplest one that explains the data.

A good model should:

1. predict unseen measurements.
2. reflect what's going on in the world.

Occam's razor says that the most parsimonious model that fits the data should provide the best predictions of future or unseen measurements. I.e. to meet the first criterion above, we should balance goodness of fit with model complexity; this is a general principle of model selection techniques. The second criterion above is more difficult to enforce. It says that the parameters we estimate should actually correspond to the physical quantity they represent. An alternative model for the exoplanet data is that the fluctuations depend on the breeding cycles of populations of giant fire birds that inhabit other suns. Periodically they all lay eggs on the surface, which

feed off the sun's energy to produce their young, thus reducing the intensity we observe remotely. Once the eggs hatch, the young firebirds fly off and the sun returns to its normal intensity. The one-firebird-population model M_1' has five parameters: the star luminance, the period and phase of the breeding cycle, the egg incubation period, and the population size; it fits the data just as well as the one-exoplanet model M_1 and provides very interesting parameters informative about the lifestyles of these exotic creatures. Is it a good model?

It is important to bare in mind that just because one model explains the data better than another, does not mean that model is right; it just explains the data better and (maybe) predicts new measurements better. Suppose in the example in figure 36 we find, through any of the model selection criteria this section covers, that M_0 explains the data best. This does not mean that the star has no orbiting planets. It simply means that our measurements do not reveal any. Wording is important when discussing and interpreting the output of a model selection procedure. We cannot say that M_1 is wrong or that M_0 is right. We can say that M_0 provides the best explanation of the data and that the data does not provide evidence for orbiting planets.

5.3 F-test for deletion of variables

The F-test is a classical hypothesis test for the inclusion or deletion of variables in a model. More broadly, the F-test tests for a difference in variance between two distributions. We can use it for model selection by testing whether the residual variance is significantly smaller for a complex model compared to a simpler model. We posit the null hypothesis that the error

variance is the same for the two models. If we can reject the null hypothesis, we choose the more complex model over the simpler.

The test relies on two assumptions: independent Gaussian noise, and nested models. Models are nested if we obtain the simpler model from the more complex one by removing some of the parameters, i.e. if the simpler model has parameters $x_1, \cdots, x_{N1}$, then the complex model has parameters $x_1, \cdots, x_{N1}, x_{N1+1}, \cdots, x_{N2}$. Models M_0, M_1, and M_2 in section 5.1 are nested models, because we can, for example, obtain M_1 from M_2 by removing all the parameters associated with the second planet. However, the diffusion tensor model and ball-and-stick model are not nested, because the simpler ball-and-stick model contains parameters that the diffusion tensor does not.

The derivation of the precise formulae underlying the F-test is somewhat involved, **NOTE 25: but see ref??**, but there is a clear and simple recipe to follow. We compute the statistic

$$F = \frac{\nu_2(\mathrm{Var}(M_2) - \mathrm{Var}(M_1))}{\nu_1 E(M_2)}, \tag{79}$$

where $\nu_1 = N_2 - N_1$, $\nu_2 = K - N_2 - 1$, K is the number of measurements, N_1 and N_2 are the numbers of parameters in models M_1 and M_2, respectively,

$$\mathrm{Var}(M) = \frac{1}{K-1}\sum_{k=1}^{K}(M(\tilde{\mathbf{x}}; \mathbf{y}_i) - \bar{M})^2 \tag{80}$$

is the variance of the model, where

$$\bar{M} = \frac{1}{K}\sum_{k=1}^{K} M(\tilde{\mathbf{x}}; \mathbf{y}_i), \tag{81}$$

$$E(M) = \frac{1}{K}\sum_{k=1}^{K}(M(\tilde{\mathbf{x}}; \mathbf{y}_i) - A_i)^2 \tag{82}$$

is the mean residual error, and $\tilde{\mathbf{x}}$ is the best-fit parameter estimates.

Under the null hypothesis, F has F-distribution with degrees of freedom ν_1 and ν_2. The F-distribution is a two-parameter distribution. The two integer parameters are called degrees of freedom and control how peaked the distribution is and how far the peak is from zero. Most statistical and numerical software packages will evaluate the F-distribution, so we can evaluate the likelihood of our F statistic under the null hypothesis and choose whether to reject the null hypothesis by comparing F to a predefined threshold.

Thus, the model selection recipe is:

- Fit both models to obtain best-fit parameter estimates.
- Compute the statistic F in Eq. 79.
- Evaluate $p = 1 - F_{\nu_1\nu_2}(F)$ (`1-fcdf(F,n1,n2)` in matlab), where $F_{\nu_1\nu_2}$ is the cumulative F-distribution function with degrees of freedom ν_1 and ν_2.
- Choose a confidence level T, such as 0.95.
- If $p < 1 - T$ reject the null hypothesis and accept the more complex model.

Consider the exoplanet example at the start of this section, where $N_1 = 5$ and $N_2 = 9$. In the first example in figure 34, $E(M_2) = 5.508 \times 10^3$, since the data set contains $K = 10000$ measurements (ten per day); $\nu_1 = 4$ and $\nu_2 = 9990$. From the fitted models, we obtain $\mathrm{Var}(M_1) = 73.1$ and $\mathrm{Var}(M_2) = 150$. Thus $F = 349$. This gives a value of p `=1-fcdf(349,4,9990)` indistinguishable from zero, very strongly suggesting that we reject the null hypothesis and accept the more complex model. In the example in figure 36, the F-statistic

comparing the zero, one and two-planet models are $F_{01} = 146$, $F_{02} = 74.6$, and $F_{12} = 2.91$; both `1-fcdf(146,4,9994)` and `1-fcdf(74.6,4,9990)` are indistinguishable from zero, but `1-fcdf(2.91,4,9990) = 0.02`, which says that we can reject the null hypothesis at the 95% significance level and accept the more complex two-planet model, but not at the 99% level.

In addition to the assumptions of independent Gaussian noise and nested models, other limitations of the F-test as a model selection criterion are:

- It can favour more complex models too strongly, particularly if the number of data points is large. **NOTE 26: Any evidence for this?**
- It does not give a direct comparative score for each model, but rather a confidence with which we can reject the null hypothesis that they are equivalent. This means that, unlike other techniques we shall come to, the F-test provides no mechanism to combine evidence from different models.

5.4 Information criteria

The next class of model selection technique computes a simple statistic that trades off fitting error and complexity (the number of parameters). The best model is the one that minimises the value of a particular information criterion of which there are several to choose from with different properties.

5.4.1 Akaike's information criterion

Akaike's information criterion (AIC) was the first statistic of this kind. The criterion is

$$AIC = 2N - 2\log L, \tag{83}$$

where N, as usual, is the number of estimated model parameters and L is the likelihood $p(\tilde{\mathbf{x}}|\mathbf{A})$ evaluated at the best guess parameter estimates $\tilde{\mathbf{x}}$. Thus if we assume independent identically distributed Gaussian noise with known standard deviation σ for all the candidate models,

$$AIC = 2N + \sigma^{-2}\sum_{k=1}^{K}(S_k - A_k)^2. \tag{84}$$

We drop the term $\log(2\pi\sigma^2)$ that arises from the normalisation constant of the Gaussian distribution, because it is independent of the choice of model. The general expression in Eq. 83 makes no assumption about the noise model, although we do need an explicit noise model to compute $\log L$. If the noise model varies among candidate models, we must be careful to retain all constant terms, as they will differ among models. If the noise standard deviation, σ, is not known a-priori, the expression for the AIC is different:

$$AIC = 2N + K\log(K^{-1}\sum_{k=1}^{K}(S_k - A_k)^2), \tag{85}$$

which is the expression in many research papers and texts on model selection, such as (Burnham and Anderson). In this situation, we use the maximum likelihood estimate of the noise standard deviation from the residual errors, which comes from

$$\tilde{\sigma}^2 = K^{-1}\sum_{k=1}^{K}(S_k - A_k)^2, \tag{86}$$

in the expression for the log likelihood:

$$\log L = \sum_{k=1}^{K}[-\frac{1}{2}\log(2\pi\tilde{\sigma}^2) - \frac{1}{2\tilde{\sigma}^2}(S_k - A_k)^2]. \tag{87}$$

If we substitute Eq. 86, the sums of squares in the second term in the sum above cancel leaving

$$\log L = -\frac{K}{2}(\log(2\pi) + \log(K^{-1}\sum_{k=1}^{K}(S_k - A_k)^2) + 1), \tag{88}$$

then dropping the constant terms $\log(2\pi) + 1$ and substituting into Eq. 83 yields Eq. 85. It is important to note that the value of N is Eq. 85 should be one greater than in Eq. 84, because in the former, we estimate one additional parameter, namely $\tilde{\sigma}$. **NOTE 27: In fact estimate of sigma should probably always be used, even if sigma is known a-priori. Check B+A.**

From Eq. 83, we can see that the AIC directly encodes the trade off between goodness of fit and model complexity. As the number of parameters increases, AIC increases, and as the goodness of fit decreases, the AIC increases. Thus the best AIC score comes from the model that maintains simplicity and goodness of fit. However, the weighting of the two terms is not arbitrary. Akaike's derivation of the AIC, which the Burnham and Anderson book explains very clearly, approximates the loss of Kullback-Liebler information

$$I(f, g) = \int f(\mathbf{y}) \frac{f(\mathbf{y}}{g(\mathbf{y}|\mathbf{x})} d\mathbf{y} \tag{89}$$

when describing the true underlying mechanism that generated the data f with each candidate model g. The criterion selects the model that minimises the expected Kullback-Liebler distance from the "truth".

Returning to the exoplanet example, we can compute the AIC of the one and two-planet models for the data in figure 34 from Eq. 85. We find that $AIC_1 = 64428$ and $AIC_2 = 63134$; the differencee of 1294 strongly favours the more complex model. For the data in figure 36, $AIC_0 = 60967$, $AIC_1 =$

60406 and $AIC_2 = 60402$, which still favours the two-planet model, but less strongly.

The standard mode of operation is to fit each model, compute the AIC for each, and choose the model with the lowest AIC score. However, we can also use the AIC to give a relative likelihood of each candidate model. The standard approach sets the relative likelihood of model i to

$$L(M_i) = \exp(\frac{AIC_{\min} - AIC_i}{2}), \tag{90}$$

where $AIC_{\min}$ is the smallest over all i. Thus we can assign a probability, or *Akaike weight*, to each model,

$$p(M_i) = \frac{L(M_i)}{\sum_j L(M_j)}, \tag{91}$$

that we can use to weight the evidence from each model in subsequent inference. For example, to estimate the value of a particular parameter that appears in each of the set of models, we can combine evidence by using the estimate that maximises

$$p(x) = \sum p(x|M_i)p(M_i). \tag{92}$$

The book by Burnham and Anderson (Model Selection and Multimodel inference, Springer-Verlag, 2002) strongly advocates this kind of treatment whenever there is uncertainty about the precise form of the model and provides several examples of its application.

The AIC in Eq. 83 is valid only in the limit as the number of measurements tends to infinity. For small sample sizes a corrected version is preferable where

$$AIC_C = AIC + \frac{2N(N+1)}{K-N-1}. \tag{93}$$

Burnham and Anderson recommend using the uncorrected AIC only when $K/N > 40$ otherwise AIC_C. The exoplanet data above has $K = 10000$ and the most complex (two-planet) model has only 9 parameters, so the uncorrected AIC is fine, but the corrected AIC is preferable in many practical situations. **NOTE 28: Provable using unbiased estimate of sigma?**

Suppose we want to predict the height of a child in a year's time. We have measurements each month over the last two years and we want to fit a model so that we can make the prediction. We are unsure which of several candidate models are best:

- Model M_0 assumes the height is constant, $h = c$.
- Model M_1 assumes linear increase with time, $h = mt + c$.
- Model M_2 assumes quadratic dependence on time, $h = at^2 + mt + c$.
- Model M_3 assumes an exponential model, $h = A\exp(mt + c) + d$

We fit each model to the available data to estimate the unknown parameters and evaluate the AIC (in fact, since the number of samples is small, we compute AIC_C instead of AIC) of each to obtain $AIC_0 = 21.5$, $AIC_1 = 13.6$, $AIC_2 = 13.8$, $AIC_3 = 15.0$. Thus, from Eq. 91, we obtain $p(M_0) = 0.008$, $p(M_1) = 0.413$, $p(M_2) = 0.374$, and $p(M_3) = 0.205$. A simple estimate of the height in the future, at say time t_f, is to take the prediction from the most likely model, M_1, i.e. $\tilde{m}t_f + \tilde{c}$. Rather than simply predict from the maximum likelihood parameter estimate, we might be more Bayesian and exploit the full posterior distribution on the parameters of M_1 to reconstruct

$$p(h|\mathbf{A}, t_f; M_1) = \int p(h|t_f, \mathbf{x}; M_1)p(\mathbf{x}|\mathbf{A}; M_1)d\mathbf{x} \tag{94}$$

and then take the maximum. However, both ignore evidence from the other models, some of which are only slightly less likely. We can incorporate the other models by integrating out the choice of model:

$$p(h|\mathbf{A}, t_f) = \sum_{i=1}^{4} p(h|\mathbf{A}, t_f; M_i)p(M_i). \tag{95}$$

A more robust choice of estimate is then the maximum of $p(h|\mathbf{A}, t_f)$. **NOTE 29: Need to make this example concrete with some actual data.**

5.4.2 Bayesian information criterion

The Bayesian information criterion

$$BIC = N \log K - 2 \log L \tag{96}$$

works in a similar way to the AIC, but with a slightly different trade off between complexity and goodness of fit. In particular, the BIC penalises complexity more as the number of measurements increase. Despite the strong similarities in structure of the two expressions, the philosophy and derivation of the BIC is quite different to that of the AIC. The BIC says nothing about the Kullback-Liebler distance between each model and the truth, which is the essence of the derivation of the AIC. The BIC is designed to pick the true model from the model set with probability 1 asymptotically as the number of samples tends to infinity; the derivation uses Bayesian arguments hence the name. A key difference in philosophy is that the design of the BIC assumes that the true model is in the model set and aims to identify it, whereas the AIC assumes that all models are approximations of the truth and aims to identify the best approximation. The latter philosophy seems more consistent with the idea of modelling and certainly more appropriate for mathematical

models in biomedicine where the underlying generating processes are way more complex than the mathematical models we use to describe them. That said, Burnham and Anderson point out that the derivation of the BIC does not rely on that basic philosophy; they provide a detailed discussion and comparison of the AIC, BIC and various other information criteria and model selection techniques.

Going back to our simulated exoplanet example, for the two-planet data in figure 34, $BIC_1 = 64527$ and $BIC_2 = 63299$, still strongly favouring the more complex model. For the data in figure 36, $BIC_0 = 61000$, $BIC_1 = 60505$ and $BIC_2 = 60567$; the stronger penalisation of complexity changes the choice of best model favouring the one-planet model over the two-planet model.

5.5 Bayesian model selection

The formal Bayesian technique for model selection aims to compute the probability of each candidate model given the data by combining evidence for each model from the data with our prior belief in each model in the usual Bayesian way. Bayes theorem helps us relate the probability of each model to quantities we can compute:

$$p(M_i|\mathbf{A}) = \frac{p(\mathbf{A}|M_i)p(M_i)}{p(\mathbf{A})}. \tag{97}$$

The prior terms $p(M_i)$ must be assigned manually through knowledge of the domain and very often we might just set them all equal to say that we have no a-priori reason to favour one model over any other. The denominator we can write as

$$p(\mathbf{A}) = \sum_i p(\mathbf{A}|M_i)p(M_i), \tag{98}$$

so the key challenge in evaluating $p(M_i|\mathbf{A})$ is to evaluate $p(\mathbf{A}|M_i)$.

At first sight, we might be tempted simply to take the ML or MAP estimate, $\tilde{\mathbf{x}}$, for each model so that $p(\mathbf{A}|M_i) = p(\mathbf{A}|\tilde{\mathbf{x}}, M_i)$. However, this favours more complex models, which fit the data more closely. For example, if all prior terms are equal and we have a Gaussian noise model, the strategy is the same as picking the model that minimises the SSD. To perform model selection properly, we need to consider all possible values of the model parameters and integrate them out:

$$p(\mathbf{A}|M_i) = \int p(\mathbf{A}|\mathbf{x}, M_i)p(\mathbf{x})d\mathbf{x}. \tag{99}$$

This step is often referred to as *marginalising out* $\mathbf{x}$; the term comes from old probability tables used pre-computers to do this kind of operation by summing over rows or columns (into the margin) of the table.

For some combinations of prior and likelihood distributions, the integral in Eq. 99 has closed form. In particular, certain combinations of choices of prior and likelihood give the same algebraic form for the posterior as the prior; these are called *conjugate priors*. For example, the Gaussian distribution is self-conjugate, so that if the likelihood and prior are both Gaussian, the posterior also has Gaussian form. There are other pairs of conjugate distributions. If any of them model the prior and likelihood well in a particular application, they are convenient choices, because they avoid lengthy numerical integration.

In general, however, the integral on the right hand side of Eq. 99 is intractable, so we have to approximate it typically using numerical sampling techniques. For low dimensional functions, regular grid sampling is plausible, but higher dimensions require other approaches. MCMC integration can be

useful, as it works for evaluating integrals of the form

$$I = \int H(\mathbf{x}) p(\mathbf{x}) d\mathbf{x} \tag{100}$$

where p is a probability distribution. We use MCMC to obtain a list of samples, $\mathbf{x}_1, \cdots, \mathbf{x}_T$, of p, and the approximation of I is simply $\sum_{i=1}^{T} H(\mathbf{x}_i)$. The marginalization integral in Eq. 99 has exactly this form with p the prior distribution and $H(\mathbf{x}) = p(\mathbf{A}|\mathbf{x}, M_i)$ the likelihood.

In summary, the recipe for Bayesian model selection is as follows:

- Evaluate the likelihood $p(\mathbf{A}|M_i)$ for each model.
- Assign a prior belief $p(M_i)$ in each model.
- Compute the posterior probability of each model from Eq. 97.
- We can then simply pick the most likely model or combine evidence from multiple models as we did with the AIC.

It is not immediately obvious how Bayesian model selection penalises complexity. The penalisation arises as a result of the integration in Eq. 99 occurring over more dimensions for more complex models. Consider a simple example of fitting a line to four data points, $A_1, \cdots, A_4$, with dependent variables $y_i = i, i = 1, \cdots, 4$; see figure 37 (left). Model M_0 assumes that the signal $S(y) = c$ is constant and independent of y; the model has one parameter, so that $\mathbf{x} = \{c\}$. Model M_1 assumes linear dependence so that $S(y) = \tan(\theta)y + c$; it has two parameters and $\mathbf{x} = \{\theta, c\}$. The data in figure 37 are actually generated from M_0 with added independent identically distributed Gaussian noise, so that $A_i = c + \eta_i$, where $\eta_i \sim N(0, \sigma)$ and

$\sigma = 0.1$. To evaluate the likelihood of M_0, we require $p(\mathbf{A}|\mathbf{x}, M_0)$. With a Gaussian noise model,

$$p(\mathbf{A}|\mathbf{x}, M_0) = \prod_{i=1}^{4} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(A_i - c)^2}{2\sigma^2}\right). \tag{101}$$

Figure 37 (right) plots $p(\mathbf{A}|\mathbf{x}, M_0)$ over the range $c \in [0, 2]$. Let's take the prior on c as uniform over that range so that the integral we need to evaluate for Eq. 99 is the integral of the function over the range shown in figure 37 (right) multiplied by the constant $p(c) = 0.5$. For model M_1,

$$p(\mathbf{A}|\mathbf{x}, M_1) = \prod_{i=1}^{4} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(A_i - \tan(\theta)y_i - c)^2}{2\sigma^2}\right). \tag{102}$$

The integral in Eq. 99 is now two dimensional, because we have two parameters. Figure 38 plots $p(\mathbf{A}|\mathbf{x}, M_1)$ over the range $c \in [0, 2]$ and $\theta \in [-\pi/2, \pi/2]$. Again assuming uniform priors over those ranges, the likelihood $p(\mathbf{A}|M_1)$ is proportional to the integral of the function in figure 38. We can see just by inspection of figures 37 and 38 that the likelihood of M_0 is higher than M_1, because the proportion of the domain over which the two-dimensional function is close to zero is much larger than that for the one-dimensional function. In fact they work out to give $p(M_0|\mathbf{A}) = 0.883$ and $p(M_1|\mathbf{A}) = 0.117$ assuming $p(M_0) = p(M_1) = 0.5$. The function in figure 37 is a slice through figure 38 at $\theta = 0$. A more direct comparison of the two likelihoods comes from extending the integral for M_0 to two dimensions by extruding that slice across the whole range of m. Figure 39 compares the two function in two dimensions from which it is simple to see that the integral for M_0 is higher. Note that the maximum of $p(\mathbf{A}|\mathbf{x}; M_1)$ is greater than that of $p(\mathbf{A}|\mathbf{x}; M_0)$, because the slice at $\theta = 0$ does not quite pass through the peak. This reflects the fact that the more complex model must always fit the data better, so that $p(\mathbf{A}|\tilde{\mathbf{x}}; M_1) > p(\mathbf{A}|\tilde{\mathbf{x}}; M_0)$.

Figures 40, 41, and 42 show a similar set of figures for data drawn from M_1 with $\theta = 14°$. The slice through $p(\mathbf{A}|\mathbf{x}, M_1)$ at $\theta = 0$, which corresponds to $p(\mathbf{A}|\mathbf{x}, M_0)$, misses the peak so has low value everywhere. In this example, $p(\mathbf{A}|\mathbf{x}, M_1)$ is about 10 orders of magnitude greater than $p(\mathbf{A}|\mathbf{x}, M_0)$ so the comparison strongly favours the two-parameter model. Note that in both examples, we have assumed knowledge of the noise standard deviation σ, whereas often in practice, we would need to estimate it from the data, which would alter the calculation. **NOTE 30: Add calculation with unknown sigma using gamma prior. NOTE 31: Use child's height example again - or something for consistency with AIC etc.**

As usual in Bayesian techniques, the results depend on the choice of prior. In the simple line-fitting example above, we can skew the model probabilities in favour of M_1 rather than M_0 by constraining the prior closely around the peak in figure 38. Although, in theory, this is a nice feature of the Bayesian approach, because it allows us to incorporate prior knowledge appropriately, in practice it is open to abuse, because we can pick priors to suit the outcome we want. Another potential criticism of the Bayesian model selection procedure is that, as discussed in section 5.4.2 on the BIC, derivations of the core formulae tend to assume that the true model is in the set of candidate models. However, this is largely a philosophical matter and the procedures should still produce sensible results in more realistic situations where we seek the best approximating model. Burnham and Anderson discuss these issues.

5.5.1 Bayes factor

A common statistic in Bayesian model selection is the Bayes factor, which is the ratio of the likelihoods of two models:

$$K = \frac{p(\mathbf{A}|M_1)}{p(\mathbf{A}|M_2)} = \frac{\int p(\mathbf{A}|\mathbf{x}, M_1)p(\mathbf{x})d\mathbf{x}}{\int p(\mathbf{A}|\mathbf{x}, M_2)p(\mathbf{x})d\mathbf{x}}. \quad (103)$$

Note that the expression does not take account of prior belief in each model, although they could potentially be used to scale each likelihood. The Bayes factor is often interpreted more loosely than the hard probabilities that formal Bayesian model selection produces. People typically say that evidence is strongly in favour of model M_1 if $K > 10$ and decisive if $K > 100$. **NOTE 32: Find a good reference for this. Mackay?**

5.6 Cross-validation

The final model selection technique we consider here is a more algorithmic approach called cross validation. The technique bypasses direct consideration of the trade off between goodness of fit and model complexity, but aims more directly to maximise the ability of the fitted model to predict unseen data. That, after all, is what motivates the trade-off in the first place.

If we are free of constraints on data acquisition and can happily acquire as much data as we like, an empirical approach provides the ultimate model selection procedure. We divide data into a training set, from which we estimate the parameters of each model we are considering, and a test set, which we use to evaluate each model's ability to predict unseen data. The model that predicts best, i.e. minimises errors on the test set, is our favoured choice and all other models are discarded.

In some situations, this is feasible. Typically, these are applications where we build a model only once and use it subsequently to analyse large amounts of future incoming data. **NOTE 33: An example from computer vision? Or machine learning?** More often, however, this is impractical. Consider a parametric mapping application in imaging for example. We have a set of candidate models and want to make a choice at each image voxel as to which model describes the local tissue and its signal the best. The full empirical approach demands at least twice the amount of data required to make a simple parameter estimate. This requires twice the imaging time, which is rarely justifiable given the budget of clinical acquisition protocols.

Cross-validation provides an approximation to this full empirical approach that is more economical in terms of data acquisition. The basic strategy is to divide the available data into training and test sets, fit on the training set and test on the test set, and repeat on many different divisions. Here are some specific strategies:

- *k-fold cross validation* randomly divides the data into k equal-sized subsets. We then use $k-1$ of the subsets combined to estimate the parameters, predict the measurements in the other subset, and compute a prediction error. The procedure repeats for each subset and computes an average error over all k subsets.

- *Repeated random subsampling* is very similar to k-fold cross validation, but draws a random sample to leave out each time so that subsets overlap, but we can examine many more combinations.

- *Leave-one-out validation* is a special case of the procedures above with $k = K-1$ so we leave out just a single measurement with each iteration.

NOTE 34: Mention jackknife

The main limitation of cross-validation in comparison to earlier model selection techniques we have seen is the computational complexity. For large numbers of complex models requiring non-linear parameter estimation, cross-validation can sometimes become prohibitively expensive in terms of computation times.

5.7 Diffusion imaging example

In diffusion imaging, the paper by Panagiotaki et al (Neuroimage 2012) defines a large number of candidate models to explain the diffusion MRI signal in white matter and uses the BIC to rank them. Later work by Ferizi et al (MICCAI 2013; Magnetic Resonance in Medicine 2014) extends the set of models and uses cross-validation to verify the ranking of models obtained from the BIC. In both cases, the set of candidate models are compartment models like the ball-and-stick model discussed in the coursework. Recall that the ball-and-stick model assumes that the signal arises from two independent contributions: *intracellular* water trapped inside axons, which can move only in the direction of the axon, and *extracellular* water outside but proximal to axons, which disperses isotropically. Panagiotaki defines several alternative models for both the intracellular and extracellular signals as well as a third compartment, which aims to model isotropic restriction arising from the presence of more spherical cells, such as glial cells, or axons with orientations different to the bulk. The full models have the form

$$S(b, \hat{\mathbf{q}}) = S_0(f_1 S_I(b, \hat{\mathbf{q}}) + f_2 S_E(b, \hat{\mathbf{q}}) + f_3 S_3(b, \hat{\mathbf{q}})), \tag{104}$$

where the volume fractions f_1, f_2 and f_3 are constrained to the range $[0, 1]$ and sum to 1.

The extracellular models each assume Gaussian dispersion via the diffusion tensor model, so that

$$S_E(b, \hat{\mathbf{q}}) = S_0 \exp(-b\hat{\mathbf{q}}^T D\hat{\mathbf{q}}), \tag{105}$$

but with different degrees of complexity in the model for D. The three extracellular models are:

- The ball model, as in the ball-and-stick model, has isotropic Gaussian dispersion, which we can represent with a diffusion tensor with three equal eigenvalues $D = dI$, where I is the identity matrix and d is the diffusivity. This model has just one free parameter, $\mathbf{x} = \{d\}$.
- The zeppelin model, in which $D = \alpha\hat{\mathbf{n}}\hat{\mathbf{n}}^T + \beta I$ has two equal eigenvalues equal to β and one unique eigenvalue equal to $\alpha + \beta$ with associated eigenvector $\hat{\mathbf{n}}$. This model has four degrees of freedom and $\mathbf{x} = \{\alpha, \beta, \hat{\mathbf{n}}\}$.
- The full diffusion tensor, which has six degrees of freedom.

The intracellular models, for S_I all assume cylindrical shaped axons with the same orientation:

- The stick model allows dispersion only in the direction of the axons, thus the radius of the cylindrical axons is effectively zero or too small to affect the measurements. The signal model fits Eq. 105 with $D = d\hat{\mathbf{n}}\hat{\mathbf{n}}^T$. The model has three degrees of freedom and $\mathbf{x} = \{d, \hat{\mathbf{n}}\}$.

- The cylinder model assumes non-zero cylinder radius R constant over all axons so has one additional parameter over the stick model with $\mathbf{x} = \{d, R, \hat{\mathbf{n}}\}$. The model does not conform to Eq. 105 and the expression is somewhat complex, so omitted here, but see (Panagiotaki et al 2012) for details.

- The gamma-distributed radii cylinder model adds one more parameter to the cylinder model to allow a distribution of cylinder radii, modelled by a two parameter gamma distribution, so that $R \sim \Gamma(k, \theta)$. The model has five degrees of freedom and $\mathbf{x} = \{d, k, \theta, \hat{\mathbf{n}}\}$.

The third compartment has four candidate models:

- The dot model assumes a compartment with negligible dispersion so negligible signal attenuation. The model has no parameters and sets $S_3(b, \hat{\mathbf{q}}) = 1$.

- The sphere model assumes spherical restriction with radius R. Again, see (Panagiotaki Neuroimage 2012) for the full expression. The model has two parameters $\mathbf{x} = \{d, R\}$.

- Astrosticks model assumes a population of axons with uniformly distributed orientation; it integrates the stick model over all $\hat{\mathbf{q}}$; see Panagiotaki for details. The only parameter is the diffusivity d.

- Astrocylinders model is similar to Astrosticks, but with non-zero cylinder radius; $\mathbf{x} = \{R, d\}$.

Figure 43 illustrates each candidate model for each compartment. The full set of candidate models then includes: 9 two compartment models, which

are each possible combination of one intracellular model and one extracellular model (no third compartment); 36 three compartment models, which are each combination of intracellular, extracellular, and third compartments from the lists above; the standard diffusion tensor model; and a two-tensor model given by

$$S(b, \hat{\mathbf{q}}) = S_0(f \exp(-b\hat{\mathbf{q}}^T D_1 \hat{\mathbf{q}}) + (1-f)\exp(-b\hat{\mathbf{q}}^T D_2 \hat{\mathbf{q}})). \quad (106)$$

The two tensor model is constrained so that both D_1 and D_2 are zeppelins, so the two-tensor model has ten degrees of freedom in total; four for each diffusion tensor, f and S_0. Various other constraints reduce the number of parameters in various compartment models, for example, in models where the extracellular compartment is anisotropic (zeppelin or tensor model), such as ZeppelinStick or TensorCylinder, the orientation of the stick or cylinder is made equal to the principal direction of the extracellular diffusion tensor. Similarly, the intrinsic diffusivity parameter in the intracellular and third compartment models is fixed equal to the principal diffusivity of the extracellular compartment: the diffusivity in the fibre direction is the same inside and outside the axons. Thus, for example, the ZeppelinCylinder model has six degrees of freedom: the five parameters of the zeppelin model, the volume fraction f, the cylinder radius, and S_0; $\hat{\mathbf{n}}$ and d in the cylinder model are set equal to the principal eigenvector and eigenvalue, respectively, of the zeppelin.

Figure 44 shows key results from the later work of Ferizi et al, which ranks the set of models according to the BIC and using cross-validation. Although rankings from the different model selection techniques show similar trends, clear differences emerge. In general in model selection, we can have more confidence in conclusions about which models are better when different kinds

of techniques give supporting results; when they give contradictory results, it is hard to draw concrete conclusions.

5.8 Summary

We have seen various techniques that encapsulate Occam's razor to determine a best choice of model from a set of candidates: the classical F-test for deletion of variables; information criteria like the AIC and BIC; and, although somewhat less directly, the Bayesian model selection mechanism. Recall from the beginning of this section that the main things we want from a model are that it makes accurate predictions of future measurements, and that it provides good estimates of unknown quantities. More empirical approaches, such as cross-validation, that go more directly for the requirement of a good model that it predicts unseen measurements. We also discussed the possibility of improving estimation of unknown quantities by combining information from multiple models using Akaike weights, Bayesian posterior model probabilities, or similar weightings derived from cross validation.

The book by Burnham and Anderson covers this topic in much greater detail. Other important considerations warrant a brief mention here. Model redundancy can skew inferences towards particular families of model that are in some way similar. Consider the extreme example of including one model twice in the set of candidates; both copies receive the same Akaike weight or posterior probability, so the model has twice the influence on subsequent inference of other models. Model redundancy can be more subtle and it is down to the designer of the set of candidate models to ensure its influence is not too great.

We must emphasise that the techniques we have discussed in this section tell us nothing about the scientific integrity of the models we choose from. Ensuring a good set of candidate models remains the domain of scientific intuition. Model selection techniques simply tell us which of the candidates explain the measurements best. We refer in places above to an abstract concept of "truth", i.e. the underlying process that actually generated the data, and reiterate the argument from Burnham and Anderson that the set of candidate models in most practical situations does not contain a true model. Model selection techniques simply aim to identify the best approximating model, but give no guarantees that even that best approximation is in any way close to the underlying "truth". Model-based techniques sometimes receive criticism for this: all the mechanisms for inference and estimation are very clever, but if the model is not correct the inference is nonsense and we never know if the model is really right. In the silly firebirds example at the start of the section, the model fits the data just as well as the exoplanets model, but is clear fantasy. The statistical techniques we have discussed do not separate them.

NOTE 35: Move somewhere else? Data-driven or phenomenological approaches that simply observe statistical differences between measurements or simple statistics derived from them are much safer, because they make no attempt to make a physical interpretation like model-based approaches do. However, the rewards are much greater from models if they are correct or even partly correct. The ability to interpret data through models really lies at the heart of what science is all about. Of course, history is littered with examples of incorrect models: the Earth-centric universe; fundamental elements of earth, wind, fire and water; Gaussian diffusion in the brain. Usually (hopefully!), they are eventually found out, because anomalies start to

appear in measurements that the models can't explain. Why do the firebirds have such consistent breeding cycles around individual stars with no fluctuation in population size and yet show such big differences between stars? These are the seed of scientific revolutions where people start to realise that a model used to explain things for maybe centuries cannot be true and radical new ways of thinking can take their place.

Figure 34: Noisy data simulated from a two-planet model.

Figure 35: Noisy data simulated from a one-planet model.

Figure 36: Noisy data simulated from an unknown model.

Figure 37: Left: data synthesized from constant model. Right: likelihoods of each data point (colours) and combined likelihood (black).

Figure 38: First four panels show the likelihood of each individual data point given each combination of θ and c, i.e. maps of $p(A_k|\theta, c)$ as a function of θ and c. The fifth panel shows the four functions summed to visualise their interaction. The final panel shows the combined likelihood of all the data $p(\mathbf{A}|\theta, c) = \prod_{k=1}^{4} p(A_k|\theta, c)$.

Figure 39: Left: $p(\mathbf{A}|\theta, c)$ as in the final panel of figure 38. Right: $p(\mathbf{A}|\theta = 0, c$ extruded to compare the functions integrated to obtain $p(M_1|\mathbf{A}$ and $p(M_0|\mathbf{A}$.

Figure 40: As figure 37 but for data synthesized from the linear model M_1.

Figure 41: As figure 38 but for data synthesized from the linear model M_1.

Figure 42: As figure 39 but for data synthesized from the linear model M_1.

Figure 43: Illustration of the Panagiotaki models from the paper by Richardson et al (Magnetic resonance in medicine 2014).

Figure 44: Results from Ferizi et al (Magnetic resonance in medicine 2014) comparing rankings of the Panagiotaki models obtained from the BIC and from cross validation.

6 Experiment design

NOTE 36: Profile Fisher?

Experiments help us learn about the world, estimate physical quantities and make predictions about the future; a similar list to that I gave earlier in section 1.2. We can summarise by saying that the purpose of an experiment is to inform a model. It might be to decide on the right model, or to estimate the parameters of a model, or to construct a model that provides predictions of the future.

Experiment design is about setting up the experiment to maximise the information we can add to our model. So far we have concentrated on the problem of estimating the parameters $\mathbf{x}$ from measurements, $A_1, \cdots, A_K$, acquired with a device settings, $\mathbf{y}_1, \cdots, \mathbf{y}_K$, respectively. The experiment design is the set of device settings $\xi = \{\mathbf{y}_1, \cdots, \mathbf{y}_K\}$. In this section, we are concerned with optimising the experiment design to get the best parameter estimates. More precisely, we might seek the choice of ξ that minimises, for example, the variance of the posterior distribution on our parameters.

6.1 General experiment design

The term "experiment design" has different, although related, meanings in difference contexts. Here we are interested in experiment design for model parameter estimation, but this section reviews some other meanings briefly to put the material we cover in context.

Classical experiment design includes problems like assigning subjects to groups

in clinical trials or, similarly, treatments to plots of land in agricultural experiments. For example, we might want to compare the effectiveness of different drugs or fertilisers. We have K participants or fields and we can apply the treatment in various strengths and/or schedules. We divide the subjects into groups and give each group a certain dose to determine the best treatment level and evaluate potential side effects. How many different groups should we use and how many different treatment levels? Two groups or K groups? Do we give any groups combinations of candidate treatments? We won't consider this kind of design here, although making these choices are well studied areas of mathematics and statistics.

Other well-studied design questions aim to maximise the chances of making the right decision. A popular example is the secretaries, or X-factor, experiment. You have a series of candidates for a job that you can interview sequentially. However, you have to make the decision yes or no at the end of each interview and cannot subsequently change your mind. The strategy that maximises the chance of finding the best candidate, is to discard all the first few candidates regardless of quality to gauge the standard, decide on the minimum acceptance criteria, then accept the first subsequent candidate to meet those criteria. It maximises the chance of getting the best candidate, but there is also a significant chance of ending up with the worst. We will not consider this kind of experiment design either.

The active learning problem in machine learning relates strongly to experiment design. The problem arises in supervised learning. Suppose we have a large number of data points that we want to train a classifier to label. We need a collection of labelled data to train the classifier. The active learning problem is to select the subset of data to label that maximises the accuracy

of the classifier trained on it. It arises when data acquisition is cheap, but data labelling is expensive. Consider, for example, automated voice transcription. It is easy to record a very large volume of speech potentially to train a system to associate words with sound. To train the system, we need that actual words that correspond to the recorded sounds. That requires manual transcription of the recordings, which is expensive. Randomly selecting a subset may not provide good coverage of the set of words: lots of repeats of some words, no examples of others. The active learning problem is to identify the least costly set of recordings to transcribe for a fully working system. Although the problem is somewhat different to ours, it uses many of the same basic tools.

Those same basic tools crop up in game theory as well, where the problem is often to make the right decision to maximise the chances of winning. There the problem is framed as seeking the set of questions that maximises knowledge of a system for a fixed number of questions. Many popular games, such as Guess Who and Mastermind, directly encode this problem. The choice of questions is an adaptive experiment design problem.

6.2 Experiment design for parameter estimation

In our context, a natural quantity to aim to minimise is the uncertainty on each unknown parameter in $\mathbf{x}$, subject to the constraint that the estimate is unbiased, i.e. over a large number of trials we get the right answer on average. Typically, we also have some constraint on the set of measurements we can acquire. As the number of measurements, or the time taken to acquire them increases, the parameter estimates generally improve as we have more

information to support them. So normally we aim to minimise the uncertainty of $\mathbf{x}$ for some fixed number of measurements. More generally, we might consider the total measurement time or cost rather than simply the number of measurements, as we may be able to acquire lots of weakly informative measurements quickly or cheaply and need to consider the option of acquiring a smaller number of more informative, but more expensive measurements instead. However, we focus on the experiment design problem with fixed K, so that the question is which K to acquire. The tools we introduce adapt easily to the broader question of minimising total measurement cost.

The strategy we take is to express the expected variance of the parameter estimates as a function of the experiment design ξ. That sets up an optimisation problem where we search for the ξ that minimises the variance of $\tilde{\mathbf{x}}$. We shall assume an unbiased estimator, as avoiding bias is more to do with finding the right fitting objective function or noise model than the right experiment design.

6.3 Fisher information

A primary tool in estimating the variance of the parameter estimates is the Fisher information matrix, which is minus the expectation of the second derivative of the log likelihood:

$$F = -E\left(\frac{\partial^2}{\partial x_i \partial x_j} \log p(\mathbf{A}|\mathbf{x})\right). \tag{107}$$

The inverse of F, evaluated at $\tilde{\mathbf{x}}$, is the expected covariance on the parameter estimates $\tilde{\mathbf{x}}$. We won't prove that here, but just give some intuition as to why this is a sensible quantity to consider in optimising experiment design. As discussed in section 4.3, the second derivative of the objective function at

the maximum likelihood estimate expresses the curvature at the peak, which is inversely proportional to the posterior variance. We want the design that minimises that variance, i.e. makes the peak of $p(\mathbf{A}|\mathbf{x})$ as sharp as possible. The expectation operator, $E(\cdot)$, arises, because we have to do that before actually acquiring any measurements so we work with the expected peak curvature given the design and noise model.

The Fisher information has closed form for some simple noise models. For example, for Gaussian noise,

$$\log p(\mathbf{A}|\mathbf{x}) = -\frac{1}{2}\sum_{k=1}^{K}\log(2\pi\sigma_k^2) - \sum_{k=1}^{K}\frac{(A_k - S_k)^2}{2\sigma_k^2}. \tag{108}$$

Thus,

$$\frac{\partial}{\partial x_i}\log p(\mathbf{A}|\mathbf{x}) = 2\sum_{k=1}^{K}\frac{\partial S_k}{\partial x_i}\frac{(A_k - S_k)}{2\sigma_k^2}, \tag{109}$$

and

$$\frac{\partial^2}{\partial x_i \partial x_j}\log p(\mathbf{A}|\mathbf{x}) = 2\sum_{k=1}^{K}\left(\frac{\partial^2 S_k}{\partial x_i \partial x_j}\frac{(A_i - S_i)}{2\sigma_k^2} - \frac{\partial S_k}{\partial x_i}\frac{\partial S_k}{\partial x_j}\frac{1}{2\sigma_k^2}\right). \tag{110}$$

The expectation of the first term on the right-hand side above must be zero, because we assume an unbiased estimator so $A_k - S_k$ is zero on average. Thus, we obtain

$$F = -E\left(\frac{\partial^2}{\partial x_i \partial x_j}\log p(\mathbf{A}|\mathbf{x})\right) = \sum_{k=1}^{K}\frac{1}{\sigma_k^2}\frac{\partial S_k}{\partial x_i}\frac{\partial S_k}{\partial x_j}. \tag{111}$$

Various standard criteria come from the Fisher information matrix to define objective functions for experiment design optimisation.

- A-optimality minimises $\mathrm{Tr}(F^{-1})$ and thus minimises the sum of the variances of all the parameters.

- D-optimality minimises $\det(F^{-}1)$ so also takes into account the covariances between each pair of parameters.
- E-optimality maximises the minimum eigenvalue of F, which minimises the maximum covariance over all linear combination of the parameters.
- T-optimality maximises $\mathrm{Tr}(F)$, which does not minimise the parameter variance directly like the other criteria, but avoids the matrix inversion so is faster to compute. It is sometimes crudely effective and, if the number of parameters is very large, may be the only practical option as an objective function.

6.4 Linear models

For a linear model,

$$S(\mathbf{y}_k) = \sum_{i=1}^{N} g_i(\mathbf{y}_k) x_i. \tag{112}$$

The first derivative

$$\frac{\partial S_k}{\partial x_i} = g_{ik} \tag{113}$$

so, from equation 111,

$$F_{ij} = \sum_{k=1}^{K} g_{ik} g_{jk}, \tag{114}$$

and the full Fisher matrix

$$F = G^T G, \tag{115}$$

where G is the design matrix, i.e. $\tilde{\mathbf{x}} = (G^T G)^{-1} G^T \mathbf{A}$. Notice that F does not depend on either the parameter values or the measurement values. Thus, by minimising F^{-1}, we minimise the covariance of the parameter estimates whatever their true values are.

Consider a simple line-fitting problem where the model is

$$h = my + c, \tag{116}$$

where the unknown parameters are $\mathbf{x} = \{m, c\}$, the measurements are of h and the sample points are the values of y for which we have measurements. The design matrix is

$$G = \begin{pmatrix} y_1 & 1 \\ \vdots & \vdots \\ y_K & 1 \end{pmatrix} \tag{117}$$

from which we can construct the Fisher information matrix

$$F = G^T G = \begin{pmatrix} \sum_{i=1}^K y_i^2 & \sum_{i=1}^K y_i \\ \sum_{i=1}^K y_i & \sum_{i=1}^K 1 \end{pmatrix} = \begin{pmatrix} \sum_{i=1}^K y_i^2 & \sum_{i=1}^K y_i \\ \sum_{i=1}^K y_i & K \end{pmatrix}. \tag{118}$$

The diagonal elements of F^{-1} are the expected variances of the unknown parameters; here $(F^{-1})_{11}$ is the expected variance of the estimate of m and $(F^{-1})_{22}$ that of c. The off-diagonal elements are expected covariances; here $(F^{-1})_{12} = (F^{-1})_{12}$ is the expected covariance of the estimates of m and c. The experiment design problem is to choose the value of $y_1, \cdots, y_K$ that give the most information about m and c. For example, for D-optimality, we minimise the determinant of F^{-1} or equivalently maximise

$$\mathrm{Det}(F) = K \left(\sum_{i=1}^K y_i^2 \right) - \left(\sum_{i=1}^K y_i \right)^2. \tag{119}$$

To illustrate, consider the simple experiment design problem $\xi = \{1, Y\}$ where we have y_1 fixed at 1, but we can vary the value of y_2 to get the best estimates of our parameters. For this 2-point ξ,

$$\mathrm{Det}(F) = (Y - 1)^2. \tag{120}$$

Thus, the determinant of F increases quadratically with Y, telling us that the D-optimality increases with Y. As we might guess, this suggests we take the second sample with y as large as possible to minimise the variance on the parameter estimates.

Now suppose the measurement variance is linear in y, i.e. the error gets bigger as $|y|$ gets bigger. Does the problem become any less trivial? Each y_i now has a unique $\sigma_i^2 = \alpha y_i$ so we have a weighted linear least squares problem. For weighted linear least squares,

$$F = G^T W G, \tag{121}$$

where, as we saw in the estimation section, W is the weights matrix, which is diagonal with $W_{ii} = \sigma_i^{-2}$. This gives us

$$F = \begin{pmatrix} \sum_{i=1}^K \frac{y_i}{\alpha} & \sum_{i=1}^K \frac{1}{\alpha} \\ \sum_{i=1}^K \frac{1}{\alpha} & \sum_{i=1}^K \frac{1}{\alpha y_i} \end{pmatrix} \tag{122}$$

and so

$$\mathrm{Det}(F) = \left(\sum_{i=1}^K \frac{y_i}{\alpha}\right)\left(\sum_{i=1}^K \frac{1}{\alpha y_i}\right) - \left(\frac{K}{\alpha}\right)^2. \tag{123}$$

For our two-point design problem, $\xi = \{1, Y\}$, this reduces to

$$\mathrm{Det}(F) = \frac{1}{\alpha^2}(Y + Y^{-1} + 2), \tag{124}$$

which again increases, but now linearly, with Y, so we still take the second sample point at the highest value of y that we can.

Finally, consider the case where the noise standard deviation increases linearly with y, so the variance is proportional to y^2. Now we have

$$F = \begin{pmatrix} \sum_{i=1}^K \frac{1}{\alpha} & \sum_{i=1}^K \frac{1}{\alpha y_i} \\ \sum_{i=1}^K \frac{1}{\alpha y_i} & \sum_{i=1}^K \frac{1}{\alpha y_i^2} \end{pmatrix} \tag{125}$$

and so

$$\mathrm{Det}(F) = \frac{K}{\alpha}\left(\sum_{i=1}^{K}\frac{1}{\alpha y_i^2}\right) - \left(\frac{K}{\alpha y_i}\right)^2. \tag{126}$$

For $\xi = \{1, Y\}$, this reduces to

$$\mathrm{Det}(F) = \frac{1}{\alpha^2}(1 - 2Y^{-1} + Y^{-2}). \tag{127}$$

This function also increases with Y although it tends to an asymptote of α^{-2}. Nevertheless, we still take the second sample point with as large a value of y as possible. If the noise is a function of any higher power of y than 2, $\mathrm{Det}(F)$ reaches a maximum at a finite value of y.

6.5 Non-linear models

For non-linear models, F depends on the parameter settings $\mathbf{x}$. For example, the non-linear ball model of the diffusion MRI signal,

$$S(b; d) = S_0 \exp(-bd), \tag{128}$$

has parameters $\mathbf{x} = \{S_0, d\}$ and first derivative

$$\frac{dS_k}{d\mathbf{x}} = \{\exp(-bd), -bS_0 \exp(-bd)\}. \tag{129}$$

Thus, assuming unbiased Gaussian noise with standard deviation σ,

$$F = \begin{pmatrix} \sum_{k=1}^{K} \exp(-2bd) & \sum_{k=1}^{K} -bS_0 \exp(-2bd) \\ \sum_{k=1}^{K} -bS_0 \exp(-2bd) & \sum_{k=1}^{K} -b^2 S_0^2 \exp(-2bd) \end{pmatrix}. \tag{130}$$

Both S_0 and d appear in the expression for F. We can still define optimality criteria based on F, such as A-optimality, D-optimality, etc, but they now depend on $\mathbf{x}$. So if we optimise any of the usual criteria, we obtain an experiment design that is optimal for only the particular choice of $\mathbf{x}$ used to

evaluation F. In practice, we often average the optimality criterion over a set of representative values of $\mathbf{x}$ and optimise an objective function like

$$V(\xi) = \sum_{q=1}^{Q} \mathrm{Tr}(F^{-1}(\mathbf{x}_q)). \tag{131}$$

6.6 Practical considerations

The sections above define the basic tools for experiment design: we construct the Fisher information matrix, derive some optimality criterion, and search for the design that optimises it. That sounds more straightforward than it is in practice, where various considerations are important to obtain good results:

- One consideration is the scale of the parameters. Consider the Fisher information matrix for the ball model in Eq. 130. The inverse matrix F^{-1} is the expected value of

$$\begin{pmatrix} \sigma_d^2 & \mathrm{Cov}(d, S_0) \\ \mathrm{Cov}(d, S_0) & \sigma_{S0}^2 \end{pmatrix}. \tag{132}$$

  If the units of d are $\mathrm{m^2 s^{-1}}$, its numerical value is of order 10^{-9} and its standard deviation is likely to be at least one order of magnitude smaller, say $\sigma_d = 10^{-10}\,\mathrm{m^2 s^{-1}}$. The units of S_0 are somewhat arbitrary, as it depends on a gain setting in the scanner, but often that might be set so that the variance of S_0 is of order 1 or in the byte or short range. Suppose we choose A-optimality and optimise $\mathrm{Tr}(F^{-1})$. The σ_{S0} term dominates the objective function, because it is many orders of magnitude larger, so we optimise the design for estimating S_0, the uninteresting nuisance parameter, without directly considering d. In

fact, in most models, the experiment design optimised for one parameter usually performs well for the others, as to estimate any parameter well, you also have to estimate the others well. However, it does make a difference to rescale the parameter values so that they have a similar influence on the objective function. One approach is to normalise F element by element by the matrix $\mathbf{x}\mathbf{x}^T$, i.e. rescale F_{ij} by $x_i x_j$.

- The choice of a-priori parameter settings is important. At first sight a Bayesian might suggest that we integrate out the parameter values to form a non-specific objective function

$$V(\xi) = \int V(\mathbf{x};\xi) p(\mathbf{x}) d\mathbf{x}. \tag{133}$$

This has the effect of weighting the most likely parameter values most strongly during the experiment design. Is it really what we want? Often the less likely parameter values are the most interesting. Suppose one in a million times we get unusual parameter settings that indicate the presence of a Higgs boson. We want to estimate those parameters at least as precisely as the more mundane values that show nothing interesting. A better strategy is to choose a set of interesting combinations that cover the likely range and weight them equally, as in Eq. 131. Results do depend on the precise choice of the $\mathbf{x}_q$ and the choice that gives the best operating performance may require considerable experimentation to determine.

- The choice of optimisation algorithm is a major consideration. Experiment design optimisation problems can be very difficult. They are often very high dimensional, in general of dimension MK, where M is the number of device settings per measurements. The objective functions from the standard criteria are often highly oscillatory so it is hard

to avoid local minima. Experiment design optimisation often requires the heavy artillery of numerical optimisation: stochastic and genetic optimisation algorithms run for days on large computer clusters.

6.7 Other approaches

A few variations on the basic experiment design problem arise frequently in practice.

- In adaptive or sequential design, we tune the acquisition on the fly to obtain the best sampling for the particular parameter values we are estimating. This only makes sense for non-linear models where the optimal design depends on the parameter values. We start by taking a few measurements, enough to make an initial parameter estimate, then choose the next one to maximise some optimality criterion given the current best guess for the parameter values. Then we can repeat the procedure until we use up our acquisition budget. Optimization is often a big challenge for adaptive design. Although each individual optimisation is less challenging than the standard problem where we optimise all sample points simultaneously, as here we optimise much fewer parameters each time (only one sample point), often the optimisation has to be done in real time to tell a device which measurement to acquire next, e.g. while a patient is inside an MRI scanner.

- We have focussed on optimising the experiment design to minimise the variance of parameter estimates. Other criteria are available for a slightly different problem, which is to optimise for the most accurate predictions of future measurements. This might, for example, be use-

ful in deciding where to place sensors used in weather or sea current predictions. Various optimality criteria for this problem are called G, I and V-optimality. It is different to maximising ξ for parameter estimation, as some parameters may affect the signal only weakly so we need less accurate estimates of those than others that are more influential on the signal.

- We can also optimise the experiment design for model selection, which again has different tactics to optimising for parameter precision. The set of measurements that distinguish two models are not necessarily the most informative about a specific parameter.

- The strategy is different again for model identification, i.e. for constructing a set of candidate models. There typically we want to explore the measurement space as widely as possible to make sure we have models that explain all variations of the signal. Or we may even want to avoid regions of the measurement space that are sensitive to a particular effect that is difficult to include in a model.

- Bayesian experiment design is somewhat less established than other areas of Bayesian parameter estimation and indirect measurement and most practical experiment design uses the methods I have already discussed. However, the Bayesians of course have some thoughts on it. I mentioned the slightly naive idea earlier of simply integrating our standard optimality criteria over the priors on the parameter values. The Bayesian experiment design formulation is more general. It defines a utility function $U(\mathbf{x}, \mathbf{y}, \xi)$ that encapsulates the aim of the experiment, e.g. to estimate certain parameters or predict future measurements.

Then the quantity we optimise is

$$U(\xi) = \int \int U(\mathbf{x}, \mathbf{y}, \xi) p(\mathbf{x}|\mathbf{y}, \xi) p(\mathbf{y}|\xi) d\mathbf{x} d\mathbf{y}. \tag{134}$$

The last term $p(\mathbf{y}|\xi)$ encapsulates uncertainty on the actual device setting $\mathbf{y}$ played out given what was instructed in ξ **NOTE 37: Check papers...**.

This way we can construct similar quantities to the various optimality criteria we have seen, but various other things are easy to construct conceptually, although often hard to evaluate in practice. One common choice for U is the maximise the Shannon entropy of the posterior, which works out similar to D-optimality:

$$U(\xi) = \int \int \log \left[p(\mathbf{x}|\mathbf{y}, \xi) \right] p(\mathbf{y}|\xi) d\mathbf{x} d\mathbf{y}. \tag{135}$$

NOTE 38: Explain how it fits the form in Eq. 134.

6.8 Example: DCE-CT

The first example comes from a short paper by Prevost et al (ISBI 2010), who set out to optimise the experiment design for dynamic contrast enhanced (DCE) computed tomography (CT) investigations used in cancer diagnosis. The procedure is to inject the patient with a contrast agent (radio tracer). The blood carries the contrast agent to the tumour and the signal gets increased in regions containing the contrast agent. When the contrast agent arrives in the tumour's blood supply, it permeates out into the body of the tumour, lingers for a while, and eventually reenters the blood and gets washed out. Thus, over time, the signal in the tumour gradually increases as the contrast agent arrives, reaches a peak and then slowly tails off. The height

and timescale of the peak depend on how easily the contrast agent gets in and out of the blood and the tissue. Blood vessels in and around tumours are leakier the faster they were generated and more malignant tumours, i.e. those that grow more quickly, tend to have leakier vessels. Thus, blood vessel leakiness correlates with cancer grade and the shape of the signal curve we measure with DCE-CT informs on leakiness, which makes the technique a useful diagnostic technique in oncology.

The signal model Prevost uses comes from the following pair of coupled differential equations:

$$\nu_1 \frac{dC_1}{dt} = \phi(C_a(t) - C_1(t)) - PS(C_1(t) - C_2(t)), \tag{136}$$

and

$$\nu_2 \frac{dC_2}{dt} = PS(C_1(t) - C_2(t)), \tag{137}$$

where ν_1 and ν_2 are the tissue volume fractions of blood and tumour body, respectively, ϕ is the flow rate of blood entering the region, PS is the permeability of the capillary walls, C_1 and C_2 are the time dependent concentrations of contrast agent in capillary blood and tumour body, respectively, and C_a is the contrast agent concentration in the arteries bringing blood to the tumour.

We can solve the system above to obtain an impulse response function

$$R(t) = A\exp(\alpha t) + (1 - A)\exp(\beta t), \tag{138}$$

for the signal at time t, where A, α and β depend on the parameters of interest $\mathbf{x} = \{\nu_1, \nu_2, \phi, PS\}$. The actual signal comes from convolving the impulse response with the arterial input function:

$$I(t) = \phi \int_0^t R(\tau) C_a(t - \tau) d\tau. \tag{139}$$

NOTE 39: Need to check Prevost for correctness and tighten.

The experiment acquires an image at each of typically $50 - 100$ time points. That provides signal intensities in each image pixel at each time point to which we fit the model in Eq. 139, either pixel by pixel or after averaging over a region of interest (a tumour), and estimate the unknown parameters, in particular PS. The main variables of the experiment design are the times at which we measure the intensity I. We might also consider the arterial input function, which we can control via the rate of injection of contrast agent, although we will assume that is fixed here.

The minimum interval between consecutive acquisitions is something like 5 or 10 seconds depending on the hardware. We could just acquire as densely as we can for a long period of time to make sure we sample the whole curve. However, the acquisitions have a cost, as each gives the patient a dose of radiation that is potentially harmful. We want to minimise the dose, but maximise the precision of PS.

Prevost reports two tests. The first optimises the position of 70 time points at least 10 seconds apart using A-optimality with parameter settings typical of a malignant tumour. He uses a stochastic population based optimisation algorithm to search for the optimal ξ. Figure 45 shows the kind of result the optimisation outputs. It shows several clusters of tightly grouped measurements with a few scattered points.

The strategy in this first optimisation has a couple of limitations. First it uses only one set of parameter values. This might be OK, if the design is tolerant to variations, but we don't know that. The second is that the optimisation is very high dimensional (70 degrees of freedom) so we cannot hope to find a

global minimum. It looks like it identifies some points of interest where the clusters form, but what about the scattered points? Are they important or just a product of not converging on the global minimum?

These limitations motivate the second test, which uses the output of the general optimisation to suggest various parametrisations of the sample points that reduce the dimensionality of the optimisation enabling the stochastic search to find better solutions. These are the parametrisations that Prevost tests:

1. Sample densely at the start and then switch to more spaced out sampling. The design parameters are: the number of sample points in each group, and the spacing of second group, assuming minimum spacing at first. The parametrisation reduces the number of degrees of freedom from 70 to 2.

2. Sample densely at the start and then use exponential spacing. The design parameters are: the number of sample points in each group, and the initial spacing and rate of increase in spacing of second group, again assuming minimum spacing for first. This parametrisation has 3 degrees of freedom.

3. The third is the fully general optimisation discussed above. This has 70 parameters, one for each individual point.

4. The last divides the measurements into four clusters of consecutive measurements together with a fifth group of exponentially spaced points. The parameters are the number of measurements in each cluster (3 degrees of freedom), the start point of each dense cluster (4 degrees of freedom) and the initial and rate of increase of spacing in the spaced

out cluster (two more degrees of freedom) for a total of 9 design parameters.

Figure 46 shows the optimal design for each parametrisation and the final value of the objective function in each case. We see that design 4 does best; it does even better than the full design because the optimisation is more constrained. The final results, in figure 47, use parametric bootstrap to assess the precision of parameter estimates using each candidate design. The results are very compelling and suggest that with the optimised designs, you can do as well as the current standard practice (design 1) with half the number of measurements if you use design 4. Either this saves the patient half the radiation dose, or you can stick with the same dose, but increase precision and thus diagnostic power significantly.

6.9 Example: Diffusion MRI

The second example continues the theme of diffusion imaging. It comes from a body of work aiming to use diffusion imaging to estimate and map the diameter of axons in white matter. The challenge is interesting in basic neuroscience, because the diameter of axons determines the speed with which they can transmit signals: larger diameters transmit information more quickly, but take up more space so cannot support the diversity of information that a larger number of small axons can. Maps of mean axon diameter over an image could support new insights into how the brain works by giving us transmission speeds of different white matter pathways. It is also a useful diagnostic parameter in certain diseases, such as multiple sclerosis and motor neurone disease, which preferentially attack axons of specific sizes.

The initial question was whether such a technique was feasible at all using current MRI hardware technology. The only way to answer such a question negatively is if we find and test the imaging protocol (i.e. the experiment design) that gives us the best chance of making such estimates. It turns out the answer to the initial question is "almost", so subsequent work took a more general approach to the experiment design problem to see if we can make significant increases in sensitivity by generalising the set of measurements we choose from.

My work in (Alexander MRM 2008) uses the zeppelin and cylinder model, using the taxonomy of Laura Panagiotaki. The model has 9 degrees of freedom in its most general form: two diffusivities and an orientation for the zeppelin, one diffusivity, an orientation and a diameter for the cylinder, and a volume fraction. This work reduced the degrees of freedom to 6 by assuming the two orientations are the same and that the cylinder diffusivity is the same as the parallel zeppelin diffusivity.

The basic measurement is the pulsed gradient spin echo sequence shown in figure 17. Each measurement has five degrees of freedom: pulse width, separation, strength and orientation. The maximum imaging time feasible routinely on healthy subjects is about 1 hour. With a standard off-the-shelf, but high end, clinical scanner, we can acquire about 360 images in that time. Thus the experiment design problem is to select the five degrees of freedom for each of those 360 images: an 1800-dimensional optimisation problem! To make this feasible, we constrain the set of orientations and divide the set of 360 measurements into M shells on N measurements. Within each shell, the measurements have evenly distributed orientation. Within each shell, the remaining degrees of freedom are just the pulse width, separation and

strength and those are the same for each measurement in the shell. The total free parameters thus reduces to $3M$. For example, we might have 4 shells of 90 directions and need to search for the 4 combinations of pulse length, strength and separation: 12 degrees of freedom in total. An initial experiment determined that indeed 4 shells of 90 is better than 3 of 120 or 5 of 72 or any more shells, insofar as it provides the minimum of the A-optimality objective function. Figure 48 shows the combination of pulse sequences in the 4-shell design for two different scenarios, one uses constraints from a clinical scanner with maximum magnetic field gradient strength $70\,\mathrm{mT\,m^{-1}}$ and the other is for a small-bore animal scanner with maximum gradient $140\,\mathrm{mT\,m^{-1}}$. A feature of both optimised designs is that they only include three unique combinations and each set of four contains a pair of identical combinations.

Subsequently, we evaluate the performance of the optimised designs by using MCMC to construct a picture of the posterior distribution on the axon diameter given measurements expected from each design. Figure 49 shows posterior distributions on axon diameters of various values for each design. The conclusion is that, while accuracy depends strongly on the available gradient strength, the technique is feasible, certainly on animal scanners and just about on human scanners. Sensitivity is weak for typical axon diameters of around one micron, but at the very least we have sensitivity to the presence or not of large axons. Two years later, we published the first axon diameter mapping techniques for both animal systems, figure 50 shows some maps over a fixed monkey brain, and live human subjects, figure 51 shows maps over a live human brain.

Subsequent work by (Drobnjak JMR 2010) generalised these ideas. The scan-

ner can manipulate the magnetic field gradient much more quickly and play out quite intricate waveforms during these pulses rather than having them fixed at one value throughout and that potentially provides much more sensitivity to the axon diameter. This subsequent optimisation describes each of four measurements as a general waveform described by a list of about 50 points constrained only by the maximum available gradient strength. The optimisation becomes very high dimensional: about 200 degrees of freedom over the four measurements. However, running it for a few days on a computer cluster gives interesting results. Combinations of square waveforms consistently emerge, as figure 52 shows. The waveforms are noisy, most likely because we cannot find the global minimum in this very high dimensional search. However, even with these suboptimal combinations, we can demonstrate dramatic improvements in estimation compared to the rectangular waveforms. Figure 53 compares posterior distributions showing that even at a maximum gradient strength of $40\,\mathrm{mT\,m^{-1}}$ we can still get good estimates of quite small axon diameters.

The two examples within the diffusion imaging theme show quite different modes of usage of the experiment design tools. In some ways, they occurred in the wrong order. The second uses the experiment design optimisation to probe a very broad potential measurement space for the kind of waveform that gives most sensitivity. It is a voyage of discovery and the analysis "discovers" these square wave oscillations. The earlier work in (Alexander MRM 2008) was much more aimed at constructing a working protocol with the best combination of settings within a more constrained set. Subsequent work (Drobnjak Micro. and Meso. Mat. 2013) following on from (Drobnjak JMR 2010) went back to a more constrained problem where the waveform is parametrized as a square wave (and various other oscillatory forms) to

construct a much more tractable problem that leads to working protocols.

6.10 Summary

A few final thoughts on experiment design that the examples above trigger:

- Often the optimal design tends towards a small number of points that are sampled repeatedly. We see this in both the DCE-CT example, where points cluster around regions of the curve, and in the axon-diameter estimation example, where we see two of the four sequence combinations repeated. This is a common feature of designs optimised in this way, particularly when they only use a single set of a-priori parameter settings, as in both of these examples. In fact, one can prove that the D-optimal solution has only as many unique points as there are parameters for just a single a-priori combination of parameter values. **NOTE 40: Find this proof...** As the prior broadens, the number of unique points increases. The uniform prior on orientation in (Alexander MRM 2008) leads to uniform directional sampling. The priors on d, R and f take a small number of representative points and average the criterion, so we obtain a small number of sample point combinations.

- Constraining the problem often gives better solutions, simply because it makes the optimisation more tractable. Even solving relatively low dimensional problems, e.g. the 12-dimensional problem in the axon diameter estimation work, is hard and requires serious optimisation machinery: stochastic population searches run on large computer clusters. Finding the global minimum of problems with several hundred degrees

of freedom is to all intents and purposes impossible. Although useful for discovering regions of the broad measurement space that are worth exploring, once we have an idea of that, imposing suitable constraints to ignore searching clearly fruitless regions helps guide the optimisation towards better solutions.

- In practice, suboptimal solutions are often good enough. We do not need to find the absolute global optimal design to improve parameter estimation performance significantly over ad-hoc designs. Of course, it is impossible to know how much better we might do by continuing to search for better and better designs.

- Experiment designs constructed in the way we discuss in this chapter assume the model is correct. That is why they tend to cluster sample points around specific parts of the measurement space where the model predicts that measurement values are most sensitive to the parameter values. They do not provide good protocols for exploring the measurement space to study the integrity of a model or explore the range of possible effects that we need to model. In those earlier stages of model development, a very different kind of experiment design is appropriate, where we sample as richly and widely as possible. The strategy is almost the opposite once we have identified a good working model: sparse and focussed acquisition in specific regions of the measurement space.

- An interesting line of research is to exploit multiple models for experiment design in a similar way to the multi-model inference advocated by the Burnham and Anderson book.

Figure 45: Prevost's optimised experiment design for 70 sampling times in dynamic contrast enhanced computed tomography.

Figure 46: Optimised parametrized designed from Prevost et al 2010.

Figure 47: Comparison of parameter estimation uncertainty from Prevost's designs compared to others.

Figure 48: Optimised pulse sequences for axon diameter estimation and mapping using diffusion MRI.

Figure 49: Posterior distributions on the axon diameter from data simulated for various true diameters.

Figure 50: Axon diameter maps over fixed monkey brain.

Figure 51: Axon diameter maps over live human brain.

Figure 52: Optimised PGSE waveforms from Drobnjak et al 2010.

Figure 53: Comparison of posterior distributions on axon diameters from PGSE protocols and general waveform protocols.

7 The modelling pipeline

We finish with a brief summary of how all the techniques we have discussed fit together in the development of a typical model-based estimation technique. I have introduced the different tools in the way that makes sense conceptually rather than the order in which they would be used in practice. Here are the steps we might typically go through during development of a model-based imaging technique:

- Construct a set of candidate models
- Acquire a rich and diverse set of measurements covering the measurement space $Y_1 \times Y_2 \times \cdots \times Y_M$, where Y_i is the range of possible settings for y_i, as widely as possible.
- Evaluate the ability of each model to explain the data using the model selection techniques in section 5.
- Identify appropriate priors on the parameter values.
- Identify any constraints on the acquisition parameters, e.g. limits imposed by the hardware or safety guidelines, or combinations of measurements that are physically unrealisable.
- Optimize the experiment design using the techniques in section 6. A two-stage process is often useful here: first, start with a very unconstrained problem to identify the broad family of measurements that are most useful for our problem; then, parametrize the problem to constrain the search to the most useful region of the measurement space to obtain a simpler optimisation problem that we can solve more easily.

- Choose a working protocol.
- Run experiments to validate the parameter estimates obtained from the estimation techniques in section 2, and evaluate confidence intervals using techniques from section 4.
- Deliver a working protocol and parameter estimation package.

Even the list above is a simplification of the development process. In practice, our understanding of the data usually develops as we construct and test new models and experiment designs. New model parameters and potential measurements emerge as we go along so the process becomes iterative rather than linear. As in the diffusion MRI experiment design example in section 6.9, practical constraints mean we might start with a simple constrained measurement space (PGSE only) rather than fully exploring the full measurement space (general waveforms).