COMPGV17: Coursework 1

Antonio Remiro Azócar, MSc Machine Learning

19 February 2017

Note 1: Figures are referenced in the report and submitted in a separate file. For information on the code, see README submitted with code listing.

Note 2: Report submitted two days late because of medical issues.

(Q1.1.1) The quality of the fit is not very good, yielding a minimised sum of squared differences of 2.880×10^7 signal units. This is expected given the naivety of the model. We compare and visualise actual measurements with our parametrised model predictions in Figure 1. Over the 108 measurements for the extracted voxel, the average measured signal is $2.352 \pm 0.938 \times 10^3$ units, whereas the average predicted signal is $2.354 \pm 0.775 \times 10^3$ units. Given a 'perfect' fit to the measured data, we would expect the measurement noise to be the only source of error. In the case of such optimal fit, the noise induces, on average, a deviation of 200 to the measured value wrt the predicted value. Hence we expect a typical value of `RESNORM` to be above $200^2 \times 108 = 4.32 \times 10^6$ units, this being a best-case sum of squared differences error. Our `RESNORM`$= 2.880 \times 10^7$ is high wrt to this value. This suggests a poor-quality fit, inducing an error nearly twice as large as the intrinsic measurement noise. The following parameter values are obtained: `S0` $= 3.511 \times 10^3$, `diff` $= -5.586 \times 10^{-6}$ mm ^{2}s^{-1}, `f` $= 1.072 \times 10^2$, `theta` $= 0.885$, `phi` $= 1.556$. Values for `diff` and `f` are not sensible since they are not physically realistic. `diff` represents the diffusivity, which must be positive, and `f` represents an anisotropic volume fraction, which must be in $[0, 1]$. The value for `S0`, the signal intensity with magnetic pulses turned off, must be positive and seems sensible. `theta` and `phi` represent rotational orientations and have sensible values.

(Q1.1.2) To allow only physically realistic settings of the parameters, the model is adapted in the following manner. Firstly, `S0` and `diff` are squared in the ball-and-stick model to ensure their positivity. `f` is transformed via the sigmoid function to be in $[0, 1]$. `theta` and `phi` do not require constraints; they represent spherical coordinates and we have $\cos(\theta) = \cos(\theta + 2\pi k)$, $\sin(\theta) = \sin(\theta + 2\pi k)$, for any integer k, for all $\theta \in \mathbb{R}$. Once the fitting is rerun, the model parameters are recovered by re-applying the transformation, in this case to the returned `parameter_hat2`. Note the inverse transformation is applied to the values in `startx` to make a 'fairer' comparison to **Q1.1.1**, recovering the same initial values for the optimisation procedure. We obtain final parameters `S0` $= 4.258 \times 10^3$, `diff` $= 0.0011$ mm ^{2}s^{-1}, `f` $= 0.357$, `theta` $= -0.985$, `phi` $= 0.579$. Evidently, these values are more realistic since `S0`, `diff` now obey positivity constraints and `f` is in $[0, 1]$. We eyeball the fit to the data by means of Figure 2, which presents a comparison of model predictions to the actual measurements. The fit appears to be of higher quality than that of **Q1.1.1**. Effectively, the new implementation yields a lower sum of squared differences, with `RESNORM` $= 5.872 \times 10^6$ units. Constraining the optimisation produces more reliable performance and provides faster convergence (25 vs. 146 iterations). This is because, in this case, we are constraining the search space close to the global minimum. With physically unrealistic parameter settings, we converge more regularly to local minima.

(Q1.1.3) To get each starting point for `S0`, `diff`, `f`, we multiply the corresponding `startx` parameter by a normally distributed number ($\mathcal{N}(0, 1)$) and add this product to the original value. For `theta` and `phi`, we add a normally distributed number multiplied by 2π to the initial zeros. Hence, the scale of each individual parameter is reflected and large enough perturbations are ensured. The parameters are later fed into the same `fminunc` constraints of **Q1.1.2**, ensuring physically realistic settings. The fitting procedure is run for 5,000 iterations; the random seed is set to three for result reproducibility. We find the solution with the smallest value of `RESNORM` (5.872×10^6 units) in 84.16% of trials. To calculate how many runs we need to be 95% sure of the global minimum, we proceed as follows. We denote the probability of obtaining the global minimum over the trials made as p; that of not obtaining it is $1 - p$. To estimate our certainty, we make use of the expression $1 - (1 - p)^n$, where n is the number of trials. For $p = 0.8416$, to be 95 % sure, $1 - 0.1584^n \geq 0.95 \rightarrow 0.1584^n \rightarrow n \log(0.1584) \geq \log 0.05 \rightarrow n \geq 1.626$. Hence we need two trials to be 95% sure of the global minimum.

(Q1.1.4) To create parameter maps for `S0`, `diff` and `f`, we must loop over each voxel dimension in the dataset. We must first compute the global minimum for each voxel. This step is carried out as in **Q1.1.3**, perturbing the initial parameters and keeping track of the minimum `RESNORM`. The fitting procedure is run for ten iterations for each voxel. Five iterations are sufficient since, under the computation in **Q1.1.3**, they correspond to being 99.99% ($1 - (1 - 0.8416)^5$) sure of the global minimum. MATLAB `try/catch` is used to handle errors, resetting the parameters to their starting value if there is an error. Every voxel iteration, parameter values and the residual error are stored into their corresponding matrix. Figures 3-6 present plots of the matrices for `S0`, `diff`, `f` and `RESNORM` respectively. These are plotted using the function `surf`. It is important to note that parameter maps over the brain slice have been plotted according to the neurological display convention. This convention corresponds to looking at the image with the patient's left on the left of the image i.e. looking at the slice from the top of the patient's head. The voxel arrays are therefore indexed spatially from left to right and from posterior to anterior. Radiological display convention, on the contrary, shows the patient's right on the left of the image.

The MATLAB function quiver is used to plot a map of the fibre direction **n**. We convert matrices of theta and phi parameters into their corresponding cartesian values via [xfib yfib zfib] = sph2cart(theta_mat, phi_mat, ones(174,145)). These values are weighted by their corresponding volume function f and fed into the quiver function (via quiver(xfib.*f_mat,yfib.*f_mat, 1.5)). The resulting plot is presented in Figure 7 ; we look at the middle of the brain slice (the lateral ventricles) more closely in Figure 8.

(Q1.1.6) This time, constrained optimization is implemented via the fmincon function. The fitting is run using the same starting parameters as in **Q1.1.1** and **Q1.1.2**. The fmincon function requires the additional inputs, A,b,Aeq, beq, nonlcon (these are set to empty arrays), lower bound lb= [0;0;0;-inf;-inf] and upper bound ub=[inf; inf; 1; inf; inf]. Note fmincon does not accept the Levenberg-Marquardt algorithm as an option; we attempt using both the active set method and sequential quadratic programming (sqp) in optimset. A table is presented in Table 1 (see Appendix), highlighting the different computation times, function counts and iterations required for different fitting algorithms. We can observe that fminunc takes the lesser computation time, 0.05s (0.07s accounting for parameter recovery, preprocessing). The global minimum is the same for all three fittings with, RESNORM $= 5.872 \times 10^6$. So are the obtained parameters S0, diff and f. For sqp, theta $= -453.4$, phi $= 6.9$; for active set, theta $= -162.4$, phi $= -109.4$.

(Q1.1.8) To account for Rician noise, we utilise an offset-Gaussian noise model with the following objective function:

$$LSE = \sum_{i=1}^{N} \frac{A_i - \sqrt{S_i^2 + \sigma^2})^2}{\sigma^2},$$

where N is the number of measurements in the signal, A is the measured signal and S is the predicted signal. σ is assumed a priori to be 200. This objective function is more computationally efficient and numerically stable than the full Rician log-likelihood objective function. In addition, it still accounts for the biases introduced by the Rician noise characteristic of MRI data.

The fitting is run using fminunc for constrained parameters. The starting point utilised in **Q1.1.2** requires perturbation; it is perturbed by adding random numbers like in **Q1.1.3**. Final parameters are not affected by the new implementation, with S0 $= 4.253 \times 10^3$, diff $= 0.0011$ mm^{2}s^{-1}, f $= 0.3581$, theta $= -8.443$, phi $= -8.846$. We eyeball the fit to the data in Figure 9; it is a higher-quality fit than those visualised in Figures 1-2 (**Q1.1.1, Q1.1.2**). The implementation yields a minimised objective function value, RESNORM $= 146.8$. Computation times, function counts and iterations required for different fitting algorithms (fminunc, fmincon-sqd, fmin-active set)) are presented in Table 2. These can be compared with the values in Table 1 (for the original objective function). Using fminunc, the new objective function significantly increases computation time (0.05 to 0.52); using fmincon, computation times are fairly similar to those in (**Q1.1.6**). This lack of increase may be attributed to the fact that using our current starting point, fmincon only converges to a local (not global) minimum (different final parameter values and RESNORM $= 206.5$). A random seed has been set for reproducibility.

(Q1.2.1) Parametric bootstrap: We first implement the procedure for voxel [65,92]. Initially, the ball and stick function from **Q1.1.1** is applied to the optimal parameters obtained for **Q1.1.2**. The computed predicted signal S is used to calculate an initial RESNORM. This value is utilised to define a constant, $k = \sqrt{\frac{\texttt{RESNORM}}{n-p}}$, where n is the number of measurements (in this case 108) and p the number of parameters (5). The starting parameters for the bootstrap procedure are set to the aforementioned optimal parameters, with the exception of setting the initial S_0 to 4.9. For each bootstrap iteration, normally distributed noise scaling with k is added to each signal element. The constrained fitting (**Q1.1.2**) is run for the modified noise and the S0, diff, f parameters are recovered and stored in lists. This procedure is repeated for 20,000 iterations. A histogram (50 bins) is plotted for each parameter and the 2σ ranges and 95% ranges are computed. Figures 10-12 present histogram counts for S0, diff and f values respectively. As expected, the plots produce Gaussian-bell-like curves, induced by the normally distributed noise. The Gaussian relationship is confirmed in two ways. Firstly, CDFs are plotted for each parameter, visualising a Gaussian fit (see Figures 13-15). Secondly, the Kolgomorov-Smirnov test is performed, yielding goodness of fit with the standard normal distribution for S0, diff and f distributions. Parametric bootstrap on voxel [65,92] results in 2σ ranges of $4.143 - 4.373 \times 10^3$ for S0, $0.0010 - 0.0012$mm^{2}s^{-1} for diff and $0.312 - 0.406$ for f. We obtain 95% ranges of $4.145 - 4.372 \times 10^3$ for S0, $0.0010 - 0.0012$mm^{2}s^{-1} for diff, and $0.312 - 0.405$ for f.

We notice that voxel [65,92] is close to the centre of the image (lateral ventricles). We repeat the previous bootstrap procedure for a voxel at the left of the brain (left of the image), [34,92]. Figures 16-18 present histogram counts for S0, diff and f values respectively, for voxel [34,92]. Again, as expected, a Gaussian relationship is observed (with the exception of a few outliers very likely caused by poorer-quality initialisation). This relationship is confirmed by CDF plots presented in Figures 19-21 and positive results of the Kolgomorov-Smirnov test. Parametric bootstrap on voxel [34,92] results in 2σ ranges of $6.676 - 6.862 \times 10^3$ for S0, $8.257 - 9.000 \times 10^{-4}$mm^{2}s^{-1} for diff and $0.141 - 0.214$ for f. We obtain 95% ranges of $6.678 - 6.859 \times 10^3$ for S0, $8.297 - 8.960 \times 10^{-4}$mm^{2}s^{-1} for diff, and $0.147 - 0.207$ for f. Parametric bootstrap on voxel [34,92] results in 2σ ranges of $6.676 - 6.862 \times 10^3$ for S0, $8.257 - 9.000 \times 10^{-4}$mm^{2}s^{-1} for diff and $0.141 - 0.214$ for f. We obtain 95% ranges of $6.678 - 6.859 \times 10^3$ for S0, $8.297 - 8.960 \times 10^{-4}$mm^{2}s^{-1} for diff, and $0.147 - 0.207$ for f. We can see that the magnitude of the ranges is very similar to that recorded for voxel

[65,92], despite the different parameter values for voxel [34,92]. Voxel [34,92] has higher `S_0` and lower diffusivity and `f` values, corresponding to its position in the brain (some gyrus).

(Q1.2.3) For the Laplace method, σ for each parameter is in the diagonal of the Hessian. The Hessian can be extracted during the model fitting procedure using `fminunc`. The 2σ range can be obtained in this way.

(Q1.2.4) To visualise the uncertainty in the fibre orientation estimate over a brain slice, we proceed as follows. We need to perform the bootstrap procedure on parameters `theta` and `phi`. In order to do this, there needs to be a trade off between computational cost and accuracy of the ranges/certainty in the global minimum/dataset size. We decide to constrict the brain slice to vertical voxels 70 to 120 and horizontal voxels 70 to 90. These correspond to the area around the brain's right ventricle. As in **Q1.1.4**, we loop over each voxel dimension, tracking the global minimum for each. The starting parameters are perturbed in the same manner, adding random values and using try/catch to handle errors. The fitting procedure is only run for two iterations for each voxel. This still gives a certainty of $(1-(1-0.8416)^2) = 97.5\%$ that we have found a global minimum. Only 100 bootstrap iterations are run for each voxel using the procedure outlined in **Q1.2.1**. `theta` and `phi` parameters are stored in lists and 2σ and 95% ranges are computed for each voxel. The lower and upper boundaries (for 2σ and 95% ranges for both `theta` and `phi`) for each voxel are stored in matrices. Once we have run through the selected voxels, `quiver` is used to plot a map of the fibre direction **n** like in **Q1.1.4**. Matrices for lower and upper boundaries of θ and ϕ are converted into their cartesian values, weighted by `f` and fed into the `quiver function`, superposing the blue fibre direction arrows. The resulting plot for the 2σ ranges is presented in Figure 22. Figure 23 presents the plot for the 95% ranges. I believe bootstrapping `theta` and `phi` this way provides a good equivalent for computing ranges in fibre direction.

(Q1.3.1) We repeat the procedure highlighted in **Q1.1.2-Q1.1.3** for the new dataset. The initial parameters need to be rescaled, corresponding to the different units of the new `bvals`. `bvals` for the ISBI 2015 dataset are in sm^{-2}, whereas those in the previous dataset are in smm^{-2}. Therefore, the diffusivity provided as a starting point for **Q1** is divided by 10^6. All other initial parameters are kept the same. We utilise the constrained model fitting implemented in **Q1.1.2**, where only physically realistic parameter settings are allowed. We obtain final parameters `S0` $= 1.010$, `diff` $= 1.431 \times 10^{-3}$ mm^{2}s^{-1}$= 1.431 \times 10^{-9}$m^{2}s^{-1}, `f` $= 0.575$, `theta` $= -1.545$, `phi` $= -0.083$. We eyeball the fit to the data by means of Figure 24, which presents a comparison of model predictions to the actual measurements. The fit appears to be of high quality , with `RESNORM` $= 15.106$. We perturb the starting points as described in **Q1.1.3** for 2000 iterations. The random seed is fixed at four for result reproducibility. With this dataset, it seems harder to find the global minimum consistently. We find the solution with the smallest value of `RESNORM` (15.106 units) in 9% of trials after making minor tweaks to our parameter generation. It is worth experimenting with different generation methods to obtain the global minimum more frequently. For $p = 0.09$, to be 95% sure, we require $1 - 0.91^n \geq 0.95 \rightarrow n \geq 31.76$ i.e. we require 32 trials to be 95% sure of the global minimum.

(Q1.3.2) Zeppelin and stick model: $\lambda_1 \geq \lambda_2$ is required and we set $\lambda_1 = 5$ and $\lambda_2 = 2$. We perturb the initial settings in **Q1.3.1** a couple of times and set different random seeds, attaining different results. A 'good-enough' result is obtained setting a random seed of 26. This fit to the data is visualised by means of Figure 25, which provides a comparison of model predictions to actual measurements. The residual error is `RESNORM` $= 46.614$. Proceeding like in **Q1.3.1**, the global minimum, `RESNORM` $= 46.614$ is obtained in 10.5% of trials, requiring 27 trials to be 95% certain. **Zeppelin and stick with tortuosity model**: λ_1 is set to 5 and $\lambda_2 = 1 - f$. The same results are obtained as with the zeppelin and stick model using the same initial parameters. **Diffusion tensor model:** An implementation of this model was attempted (see code). In general, I believe a greater effort should be made for a more robust, numerically stable, parameter generation process. Diffusivities of the 10^{-9}mm^{2}s^{-1} order may have triggered underflow problems and working in log space could be a better idea.

(Q1.3.3) Bayesian Information Criterion: The BIC for all three models is calculated using the following equation:

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

where K is the number of data points (in this case 3612), N is one greater than the number of model parameters (in this case 6, accounting for the additional parameter σ - noise standard deviation). **Akaike Information Criterion**: The AIC for all three models is calculated using the following (non-corrected) equation:

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

Note the final sum in the expression corresponds to the SSD (residual error `RESNORM`) computation so both calculations are easily integrated in our functions. For the ball and stick model, $AIC = -1.977 \times 10^4$, $BIC = -1.973 \times 10^4$. For the zeppelin and stick model, and for the zeppelin and stick model with tortuosity, $AIC = -1.570 \times 10^4$ and $BIC = -1.566 \times 10^4$. The lower information criteria values given by the ball and stick model make it the 'preferred model'-the model giving the better fit (from a statistical perspective, reflecting a better trade-off between the lack of fit and the number of parameters in the model).