

## Chapter 1

# Fitting Non Linear Regression Models With the Parameter Cascade

Consider the following non-linear regression model where observed values  $y_i$  are values observed at times  $t_i$  :

$$y_i = \alpha + \beta e^{\gamma t_i} + \epsilon_i \quad (1.1)$$

All of the difficulty here comes from the  $e^{\gamma t}$  term. If  $\gamma$  were known,  $\alpha$  and  $\beta$  could be found through simple linear regression with the  $e^{\gamma t_i}$  term acting as an independent variable predicting the  $y_i$

This suggests the following regression strategy. Define a function  $H(\gamma)$  to be the sum of squared errors from performing simple linear regression on the  $y_i$  against  $e^{\gamma t_i}$ . That is:

$$H(\gamma) = \min_{\alpha, \beta} \sum [y_i - \alpha - \beta e^{\gamma t_i}]^2$$

This defines a middle problem, with the inner problem being that of minimising the simple linear regression problem given  $\gamma$ . The non-linear model can be fitted by using Brent's Method to fit the middle problem.

This approach was applied to simulated data with  $\alpha = 100, \beta = 4$ , and  $\gamma = 1$ , and the results can be seen in Figure 1.1.

## 1.1 Fitting Linear Homogenous ODEs Using the Parameter Cascade

Recall a linear homogenous ODE of order  $n$  is given by:

$$\frac{d^n y}{dt^n} = \sum_{k=0}^{n-1} a_k(t; \theta) \frac{d^k y}{dt^k}$$

Under some mild technical conditions, the set of solutions to such an ODE is an  $n$  dimensional vector space and has a unique solution for each set of initial conditions. It is often more convenient to work with ODEs in matrix form from now on. Any homogenous linear ODE can be represented in matrix form:

$$\frac{d\mathbf{y}}{dt} = \mathbf{A}(t; \theta)\mathbf{y}$$

For example, the ODE  $y'' = -\omega^2 y$  with the initial conditions  $y(0) = y_0$  and  $y'(0) = v_0$  can be represented as:

$$\frac{d}{dt} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix} = \begin{pmatrix} 0 & 1 \\ -\omega^2 & 0 \end{pmatrix} \begin{pmatrix} y_1 \\ y_2 \end{pmatrix}$$

with the initial condition  $\mathbf{y}(0) = (y_0, v_0)'$ .

A basis for the solution set of any linear ODE can be formed from the set of solutions associated with the initial conditions  $\mathbf{y}(0) = \mathbf{e}_i$ , where  $\mathbf{e}_i$  denotes the  $i$ th basis vector.

So this suggests the following cascade algorithm: given a set of parameters, find the set of solutions  $\{y_1(t|\theta), \dots, y_n(t|\theta)\}$ , where  $y_k(t|\theta)$  denotes the solution with the  $k$ th basis vector as an initial condition. Then perform regression to fit the  $\{y_i(t|\theta)\}$  to the observed data. The inner problem consists of fitting a weighted sum of the  $\{y_i(t|\theta)\}$  to the observed data and reporting the associated error given a choice of parameters. The middle problem consists of finding the set of parameters that minimises this associated error.

For a problem where the ODE can be solved explicitly, things proceed as in the previous section. Consider again the ODE  $y'' - \omega^2 y = 0$ . The solutions generated by the initial conditions  $(1, 0)$  and  $(0, 1)$  is given by  $\{A \cos \omega t + B \sin \omega t | A, B \in \mathbb{R}\}$ . So the middle least squares criterion is given by:

$$H(\omega) = \min_{a,b} \sum [y_i - a \cos \omega t_i - b \sin \omega t_i]^2$$

Finding the optimal  $a$  and  $b$  given  $\omega$  is an inner problem that can be solved using least squares regression as before. In fact, such a problem has already been encountered: the nonlinear model given in Equation 1.1 is associated with the ODE  $y'' - \gamma y' = 0$ .

For ODE problems that cannot be explicitly solved, the trajectories  $y_n(t|\theta)$  must be instead found by a numerical solver for each choice of  $\theta$ . The inner problem then consists of linear regressing the computed solutions against the observed data.

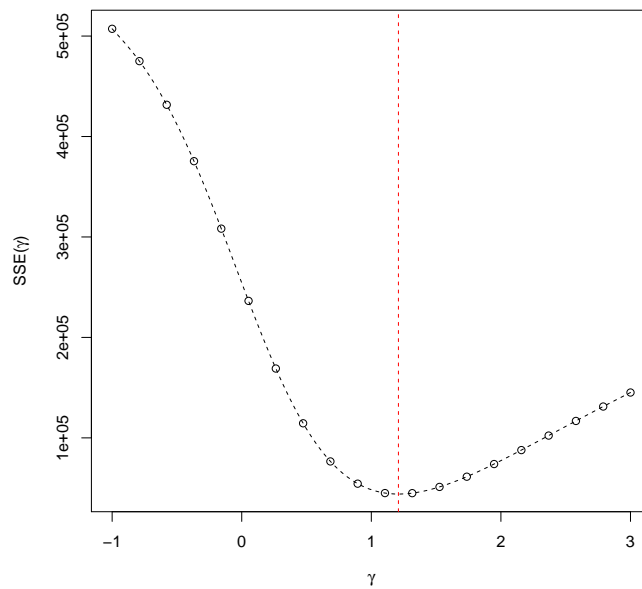
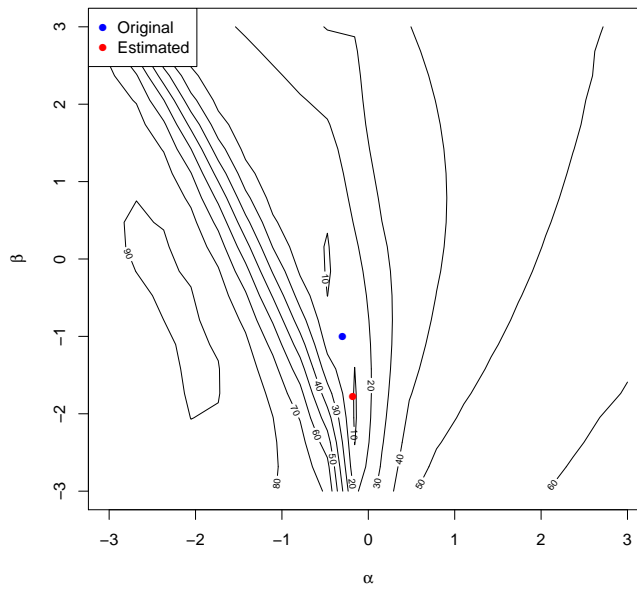
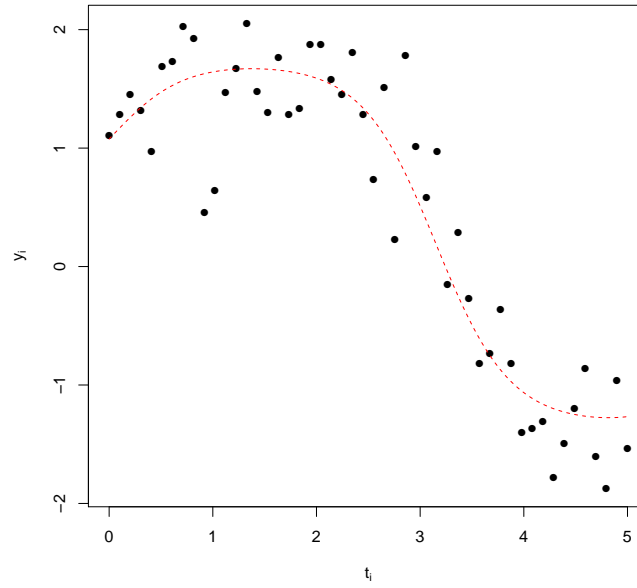


Figure 1.1: Profile Plot and Fitted Curve

To illustrate the method, it was applied to the following ODE with  $\alpha = -0.3$  and  $\beta = -1.0$ . To minimise the middle problem, the Nelder-Mead method was used - Brent's method was felt to be unsuitable because of the awkward topography. The results can be seen in Figure 1.2.

$$y''(t) = \alpha\sqrt{t}y(t) + \beta\sin(2t)y'(t) \quad (1.2)$$

The advantage of the parameter cascade here is that it is noticeably faster than trying to optimise everything in one go. The linear regression steps mean that the ODE needs to be solved numerically fewer times, so that the algorithm runs around 30% faster for the ODE in Equation 1.2. However, the `nls` command is faster than the Parameter Cascade for Equation 1.1 even when no derivatives are provided.



## 1.2 Estimation for First Order Linear PDEs

As discussed in Chapter 2, a similar framework can be used to perform estimation for PDEs in some cases. A complication is that for a PDE, the initial condition is a function rather than a constant.

### 1.2.1 The Transport Equation

In Chapter 2, the Transport equation was introduced. The Transport Equation is defined by:

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} + \beta x \frac{\partial u(x,t)}{\partial t} &= 0 \\ u(x, 0) &= f(x) \end{cases} \quad (1.3)$$

In Chapter 2, we used a middle objective function  $H(\beta)$  to estimate the parameter  $\beta$  was defined, but no effort was made to actually fit the model.

Recall that  $H(\beta)$  was defined as the sum of squares:

$$H(\beta) = \sum [y_i - \hat{f}(x_i - \beta t_i)]^2$$

And so:

$$\frac{\partial H}{\partial \beta} = - \sum 2t_i \hat{f}'(x_i - \beta t_i) [y_i - \hat{f}(x_i - \beta t_i)]$$

To compute the gradient of  $H(\beta)$ , the estimates of the functions  $f(x)$  and  $f'(x)$  associated with a given choice of  $\beta$  are needed.

This understates the difficulty however. The command `smooth.spline` will only return the GCV score, not the sum of squared errors. So we are forced to use a more complicated objective function than least squares unless a routine to compute them is written.

Fortunately, Brent's Method can be used to minimise  $H(\beta)$  instead. As can be seen in Figure 1.3 objective function is irregular though, and care must be taken that one is close to the optimal value already.

Estimating  $f(x)$  is harder than estimating  $\beta$  as can be seen in Figure 1.4.

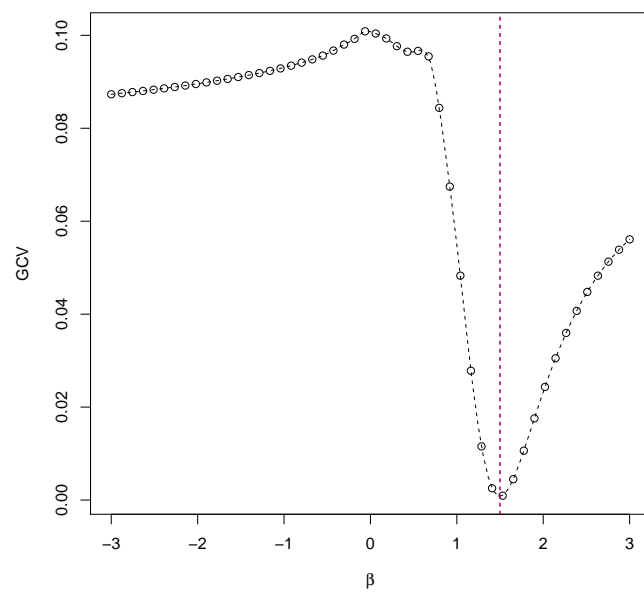
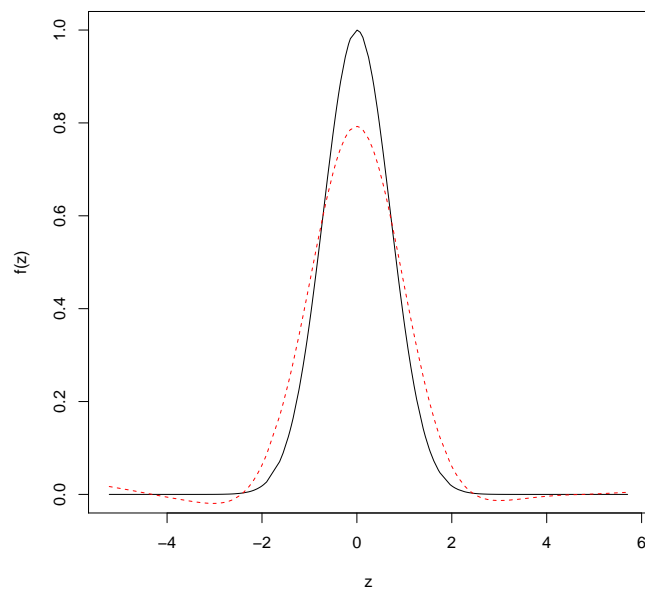
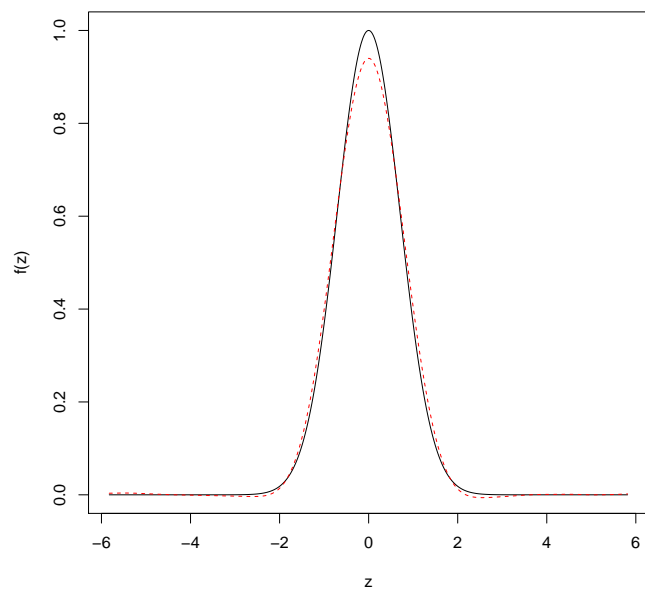


Figure 1.3: Plot of outer optimisation for the transport equation. Blue line denotes original parameter, red line denotes fitted estimate



(a)



(b)

Figure 1.4: The estimates of  $f(x)$  computed for various sample sizes. The plot (a) gives the result with 200 sample points, the plot (b) gives the result for 2000 sample points. To avoid confusion between  $x$  and  $x - \beta t$ ,  $z$  was made the independent variable for this plot. Furthermore, the points used to fit  $f(x)$  aren't displayed to reduce clutter and make it easier to compare the fitted curves with the original.



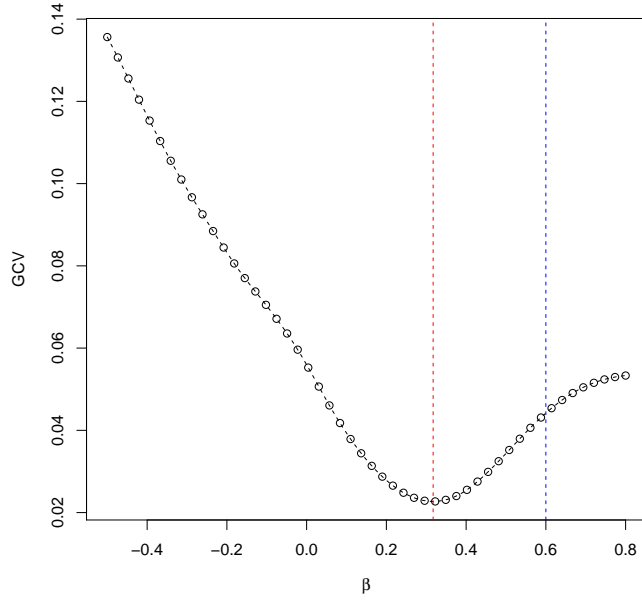


Figure 1.5: Plot of outer optimisation for the modified transport equation with only 30 sample points. Blue line denotes original parameter, red line denotes fitted estimate

### 1.2.2 A Less Trivial PDE

The transport equation is a little trivial, so the methodology will be applied to a more difficult PDE. Instead of having a constant velocity, let it vary with position by having the velocity equal  $\beta x$  instead of  $\beta$ . This produces the following modified transport equation:

$$\begin{cases} \frac{\partial u(x,t)}{\partial t} + \beta x \frac{\partial u(x,t)}{\partial t} &= 0 \\ u(x, 0) &= f(x) \end{cases} \quad (1.4)$$

The problem of estimating  $\beta$  for this PDE is ill-conditioned in the sense that `smooth.spline` will crash for some meshes.

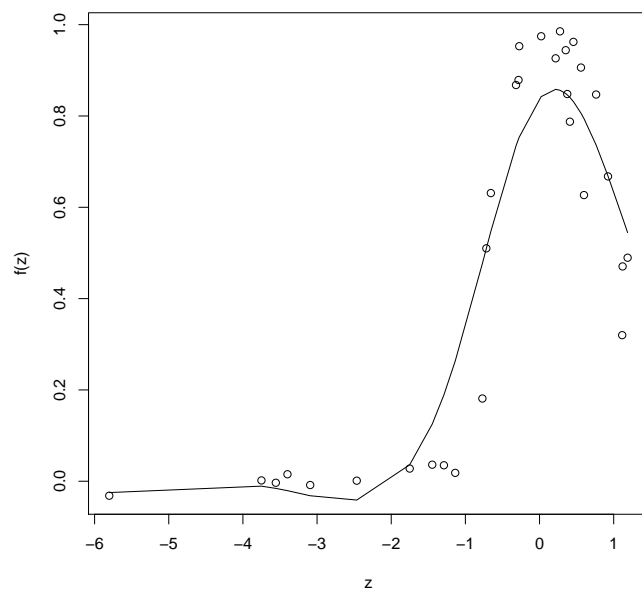


Figure 1.6: Plotted estimate of  $f(x)$  for the modified transport equation with 30 sample points.