19

Fitting differential equations to functional data: Principal differential analysis

19.1 Introduction

Now that we have fastened a belt of tools around our waists for tinkering with differential equations, we return to the problems introduced in Chapter 17 ready to get down to some serious work.

Using a differential equation as a modelling object involves concepts drawn from both the functional linear model and from principal components analysis. A differential equation can certainly capture the shape features in both the curve and its derivative for a single functional datum such as the oil refinery observation shown in Figure 1.4. But because the set of solutions to a differential equation is an entire function space, it can also model variation across observations when $N > 1$. In this sense, it also has the flavor of principal components analysis where we find a subspace of functions able to capture the dominant modes of variation in the data.

We have, then, a question of emphasis or perspective. On one hand, the data analyst may want to capture important features of the dynamics of a single observation, and thus look *within* the m-dimensional space of solutions of an estimated equation to find that which gives the best account of the data. On the other hand, the goal may be to see how much functional variation can be explained *across* multiple realizations of a process. Thus, linear modelling and variance decomposition merge into one analysis in this environment.

We introduce a new term here: *principal differential analysis* means the fitting of a differential equation to noisy data so as to capture either the

features of a single curve or the variation in features across curves. This term was first used in Ramsay (1996a), and will be motivated in some detail in Section 19.6. The abbreviation PDA will be handy, and will also serve to remind us of the close connection with PCA.

19.2 Defining the problem

Our challenge is the identification of a linear differential operator

$$L = \beta_0 I + \ldots + \beta_{m-1} D^{m-1} + D^m \tag{19.1}$$

and its associated homogeneous differential equation

$$D^m x = -\beta_0 x - \ldots - \beta_{m-1} D^{m-1} x \tag{19.2}$$

using a set of N functional observations x_i along with, possibly, a set of associated functional covariates $f_{i\ell}, \ell = 1, \ldots, L$. We now call these covariates *forcing functions* so as to keep the nomenclature already current in fields such as engineering and physics. Although, in the examples used in this chapter, the x_i's are univariate functions, and only one forcing function, if at all, is used, we certainly have in mind that systems of differential equations and multiple forcing functions may be involved, and the differential equations may be nonlinear.

First, consider the homogeneous case, where no forcing function is present. We want to find the operator L that comes as close as possible to satisfying the homogeneous linear differential equation

$$Lx_i = 0, \ i = 1, \ldots, N. \tag{19.3}$$

In order to achieve this, we have to estimate up to m coefficient functional parameters $\beta_j, j = 0, \ldots, m-1$. Of course, some of these parameters may be fixed, often to zero as we have already seen, and the constant coefficient case is included within this framework by using a constant basis where required.

Since we wish the operator L to annihilate as nearly as possible the given data functions x_i, we regard the function Lx_i as being the residual from the fit provided by the corresponding linear differential equation (19.2). The least squares approach defines as the fitting criterion the sum of squared norms of the residual functions Lx_i:

$$\texttt{SSE}_{\text{PDA}}(L|\mathbf{x}) = \sum_{i=1}^{N} \int [Lx_i(t)]^2 \, dt = \sum_{i=1}^{N} \|Lx_i\|^2. \tag{19.4}$$

If an input forcing function f_i has also been observed along with the output x_i for a system, then we aim to solve as closely as possible the nonhomogeneous equation

$$Lx_i = f_i, i = 1, \ldots, N.$$

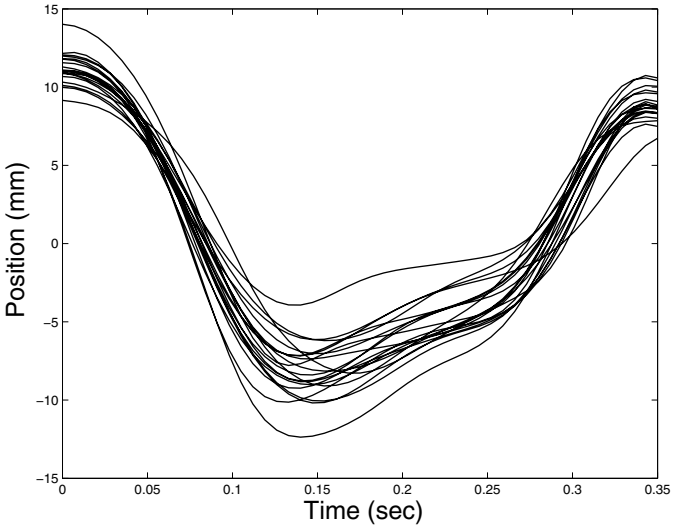

Figure 19.1. Twenty records of position of the center of the lower lip during the uttering of the syllable "bob."

The least squares fitting criterion now becomes

$$\mathtt{SSE}_{\mathrm{PDA}}(L|\mathbf{x},\mathbf{f}) = \sum_{i=1}^{N} \int [Lx_i(t) - f_i(t)]^2 \, dt = \sum_{i=1}^{N} \|Lx_i - f_i\|^2 \qquad (19.5)$$

It will be evident, when we compare these criteria with those for the concurrent functional linear model (14.5), that we may use the same methods here. Indeed, that is what we did in Chapter 17 for the oil refinery and melanoma data. However, there are other estimation techniques available that may be better. But before we consider these, we offer two examples to illustrate some of the issues involved in PDA.

19.3 A principal differential analysis of lip movement

There are several reasons why a PDA can provide important information about the data and the phenomenon under study. Certainly, in many applications the differential equation $Lx = 0$ offers an interesting and useful way of understanding the processes that generated the data.

Consider as an example to be used throughout this chapter the curves presented in Figure 19.1. These indicate the movement of the center of the lower lip as a single speaker said "bob." The displayed curves are the result of considerable preprocessing, including smoothing and the use of

functional PCA to identify the direction in which most of the motion was found. Details can be found in Ramsay, Munhall, Gracco and Ostry (1996). We see in broad terms that lower lip motion shows three phases: an initial rapid opening, a sharp transition to a relatively slow and nearly linear motion, and a final rapid closure.

19.3.1 The biomechanics of lip movement

Because the lower lip is part of a mechanical system, inevitably having certain natural resonating frequencies and a stiffness or resistance to movement, it seems appropriate to explore to what extent this motion can be expressed in terms of a second order linear differential equation of the type useful in the analysis of such systems,

$$Lx_i = \beta_0 x_i + \beta_1 Dx_i + D^2 x_i = 0. \tag{19.6}$$

Discussions of second order mechanical systems can be found in most applied texts on ordinary differential equations, such as Tenenbaum and Pollard (1963).

The first coefficient, β_0, essentially reflects the position-dependent force applied to the system at position x. Coefficient values $\beta_0 > 0$ and $\beta_1 = 0$ correspond to a system with sinusoidal or harmonic motion, with $\beta_0^{1/2}/(2\pi)$ cycles per unit time and wavelength or period $2\pi\beta_0^{-1/2}$; β_0 is often called the *spring constant*. The second coefficient, β_1, indicates influences on the system that are proportional to velocity rather than position, and are often internal or external frictional forces or viscosity in mechanical systems.

The *discriminant* of the second order operator and the mechanical system that it represents is defined as $d = (\beta_1/2)^2 - \beta_0$, and is critical in terms of its sign. When β_1 is small, meaning that d is negative, the system is underdamped, and tends to exhibit some oscillation that gradually disappears. When d is positive because β_1 is relatively large, the system is called overdamped, and either becomes stable so quickly that no oscillation is observed ($\beta_1 > 0$), or oscillates out of control ($\beta_1 < 0$). A critically damped system is one for which $d = 0$, and it exhibits non-oscillatory motion that decays rapidly to zero.

These mechanical interpretations of the roles of coefficient functions β_0 and β_1 are, strictly speaking, only appropriate if these functions are constants, but higher-order effects can be ignored if they do not vary too rapidly with t, in which case β_0, β_1, and d can be viewed as describing the instantaneous state of the system. When $\beta_0 = \beta_1 = 0$ the system is in linear motion, for which $D^2 x = 0$.

The techniques we develop were used to obtain the weight functions displayed in Figure 19.2. These are of rather limited help in interpreting the system, but one does note that β_0 is positive except for a brief episode near the beginning, and near zero in the central portion corresponding to the

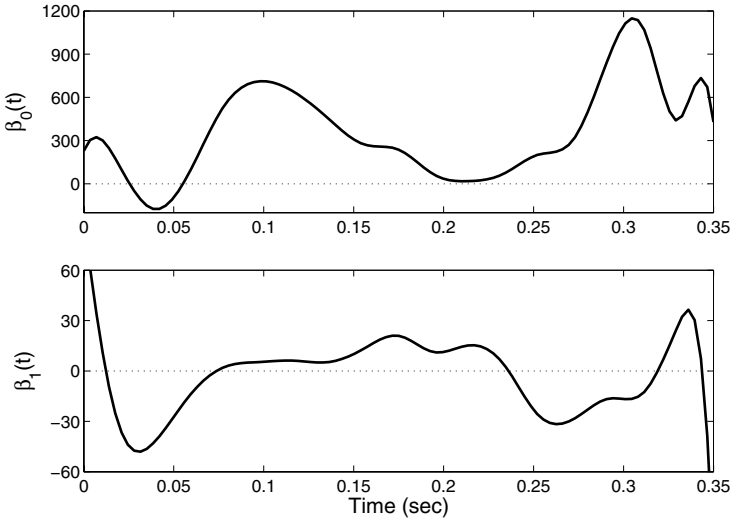

Figure 19.2. The two weight functions β_0 and β_1 for the second order linear differential equation estimated from the lip movement data.

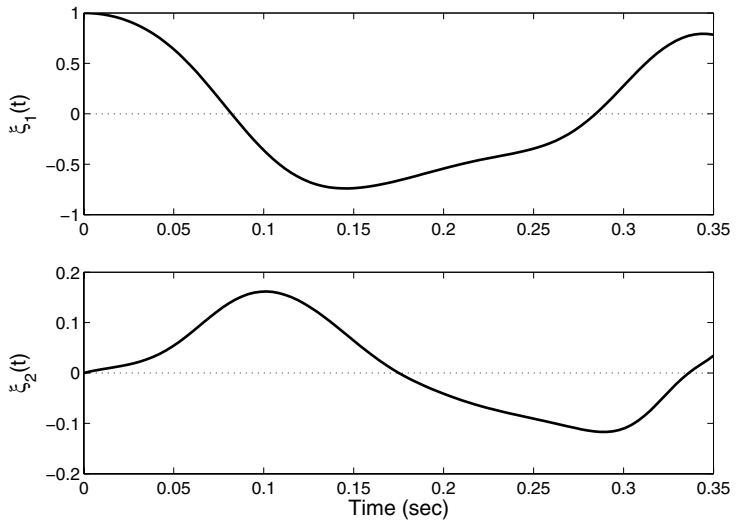

Figure 19.3. Two solutions of the second order linear differential equation estimated for the lip movement data corresponding to initial values conditions $(x(0) = 1, Dx(0) = 0)$ and $(x(0) = 0, Dx(0) = 1)$.

near linear phase of lip movement. The two solutions to the homogeneous differential equation $Lu = 0$ defined by the initial value conditions $(x(0) = 1, Dx(0) = 0)$ and $(x(0) = 0, Dx(0) = 1)$ are shown in Figure 19.3.

19.3.2 Visualizing the PDA results

How effective is the differential operator L at annihilating variation in the x_i? We can see this by plotting the *empirical forcing functions* Lx_i. If these are small and mainly noise-like, we can have some confidence that the equation is doing a good job of representing the data. It is easier to see how successful we have been if we have a null or benchmark hypothesis. A reasonable choice is the model defined by $\beta_0 = \ldots = \beta_{m-1} = 0$. The $D^m x_i$'s are the empirical forcing functions corresponding to this null hypothesis, and we can therefore compare the size of the Lx_i's to these derivatives.

Figure 19.4 shows the acceleration curves for the lip data in the left panel, and the empirical forcing functions in the right. We see that the forcing functions corresponding to L are indeed much smaller in magnitude, and more or less noise-like except for two bursts of signal near the beginning and end of the time interval.

The value of the criterion $\texttt{SSE}_{\text{PDA}}$ defined above is 7.7×10^6, while the same measure of the size of $D^2 x_i$'s is 90.4×10^6. If we call the latter measure $\texttt{SSY}_{\text{PDA}}$, then we can also summarize these results in the squared correlation measure

$$\texttt{RSQ}_{\text{PDA}} = (\texttt{SSY}_{\text{PDA}} - \texttt{SSE}_{\text{PDA}})/\texttt{SSY}_{\text{PDA}}, \tag{19.7}$$

the value of which is 0.92 for this problem.

While it is strictly speaking not the task of PDA to approximate the original curves (this would be a job for PCA), we can nevertheless wonder how well the two solution curves would serve this purpose. Figure 19.5 shows the least squares approximation of the first two curves in terms of the two solution functions in Figure 19.3, and we see that the fit is fairly satisfactory.

Finally, we return to the discriminant function $d = (\beta_1/2)^2 - \beta_0$, presented in Figure 19.6, and its interpretation. This system is more or less critically damped over the interval $0.18 \leq t \leq 0.26$, suggesting that its behavior may be under external control. But in the vicinities of $t = 0.08$ and $t = 0.30$, the system is substantially under-damped, and thus behaving locally like a spring. The period of the spring would be around 30 to 40 msec, and this is in the range of values estimated in studies of the mechanical properties of flaccid soft tissue. These results suggest that the external input to lip motions tends to be concentrated in the brief period near $t = 0.20$, when the natural tendency for the lip to close is retarded in order to allow for the articulation of the vowel.

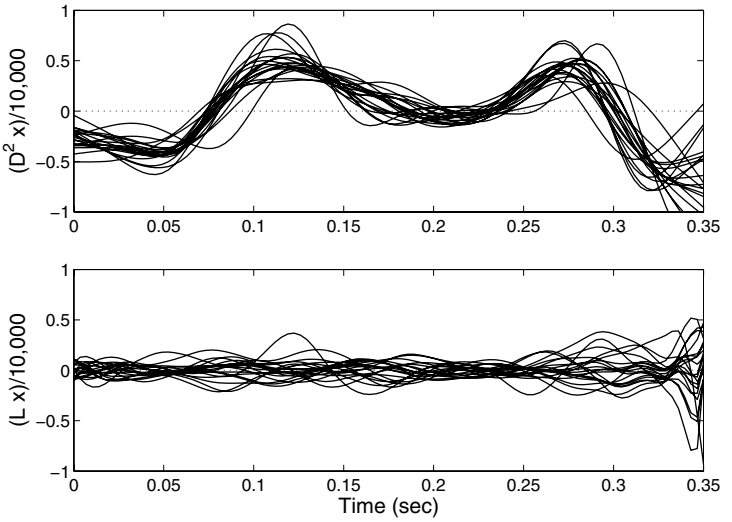

Figure 19.4. The left panel displays the acceleration curves D^2x_i for the lip position data, and the right panel the forcing functions Lx_i.

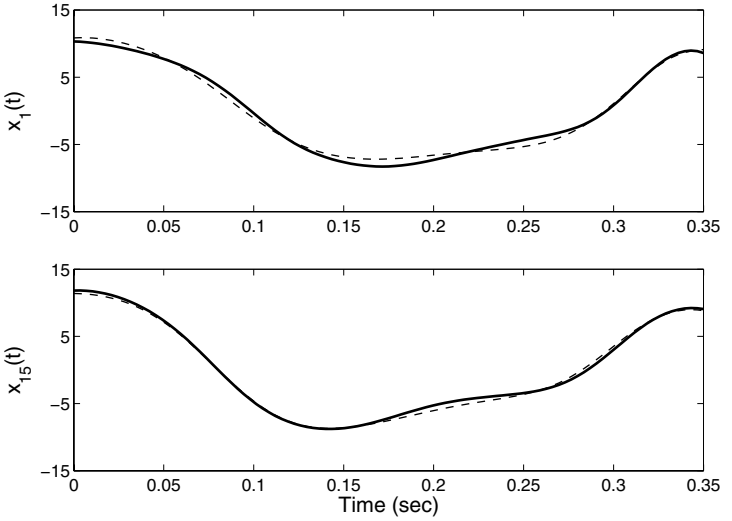

Figure 19.5. The solid curves are the first two observed lip position functions, and the dashed lines are their approximations on the basis of the two solution functions in Figure 19.3.

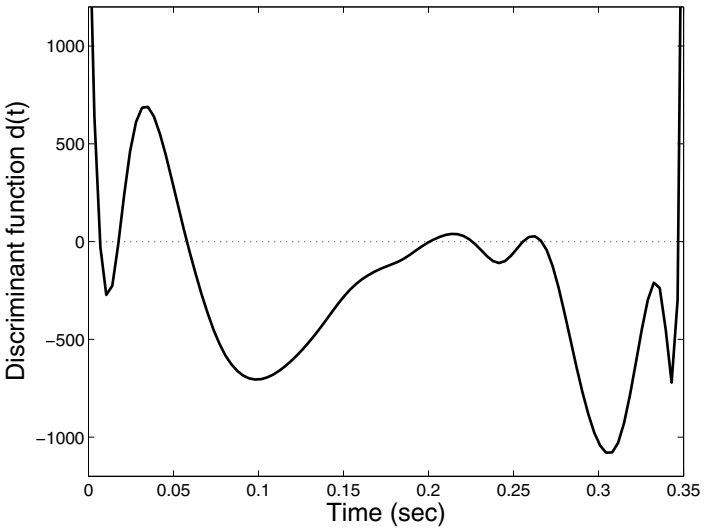

Figure 19.6. The discriminant function $d = (\beta_1/2)^2 - \beta_0$ for the second order differential equation describing lip position.

19.4 PDA of the pinch force data

In this section we take up an example in which the estimated linear differential operator is compared with a theoretically defined operator. The data in this example consisted of the 20 records of brief force impulses exerted by the thumb and forefinger in the experiment in motor physiology described in Section 1.5.2. For the purposes of this discussion, the force impulses were preprocessed to transform time linearly to a common metric, and to remove some simple shape variation. The resulting curves are displayed in Figure 19.7. Details concerning the preprocessing stages can be found in Ramsay, Wang and Flanagan (1995).

There are some theoretical considerations which suggest that the model

$$y_i(t) = C_i \exp[-\log^2 t/(2\sigma^2)] \tag{19.8}$$

offers a good account of any specific force function. In this application, the data were preprocessed to conform to a fixed shape parameter σ^2 of 0.05. Functions of the form (19.8) are annihilated by the differential operator $L_0 = [(t\sigma)^{-1} \log t]I + D$. A goal of this analysis is to compare this theoretical operator with the first order differential operator $L = \beta_0 I + D$ estimated from the data, or to compare the theoretical weight function $\omega_0(t) = (t\sigma)^{-1} \log t$ with its empirical counterpart β_0.

We smoothed the records using splines, with the size of the third derivative being penalized in order to get a smooth first derivative estimate. It is clear from Figure 19.7 that the size of error variation is not constant over

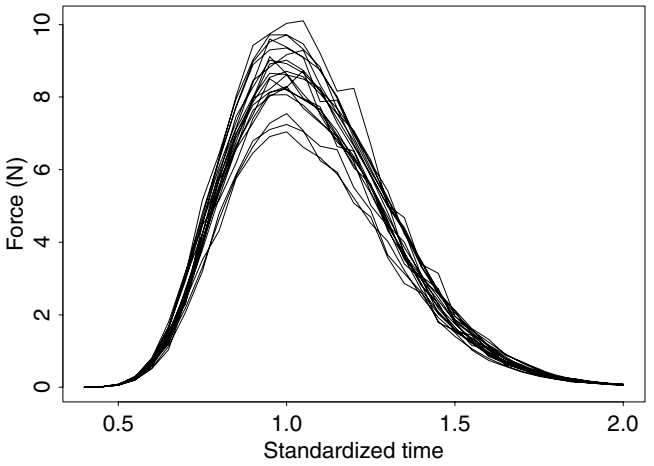

Figure 19.7. Twenty recordings of the force exerted by the thumb and forefinger during a brief squeeze of a force meter. The data have been preprocessed to register the functions and remove some shape variability, and the values displayed are for the 33 values $t = 0.4(.05)2.0$.

time. Accordingly, we estimated the residuals in a first smoothing step, and smoothed the logs of their standard deviations to estimate the variation of the typical residual size over time. We then took the weights σ_j^2 in the weighted spline smoothing criterion

$$\texttt{PENSSE}_\lambda(\mathbf{x}|\mathbf{y}) = \sum_j [y_j - x(t_j)]^2/\sigma_j^2 + \lambda \|D^3 x\|^2 \tag{19.9}$$

to be the squares of the exponential-transformed smooth values. Finally, we re-smoothed the data to get the spline smoothing curves and their derivatives. Figure 19.8 displays the discrete data points, the smoothing function, and also the theoretical function (19.8) fit by least squares for a single record. The theoretical function fits very well, but in the right panel we see that the discrepancy between the theoretical model and the smoothing spline fit is nevertheless smooth and of the order of the largest deviations of the points from this flexible spline fit. While this discordance between the model and the spline is less than 2% of the size of the force itself, we are nevertheless entitled to wonder if this theoretical model can be improved.

We applied both the point-wise and basis expansion procedures for estimating β_0 to the smooth functions and their derivatives, as described in Section 19.5. The basis used for the basis expansion procedure was

$$\phi(t) = (t^{-1}\log t, 1, t-1, (t-1)^2)',$$

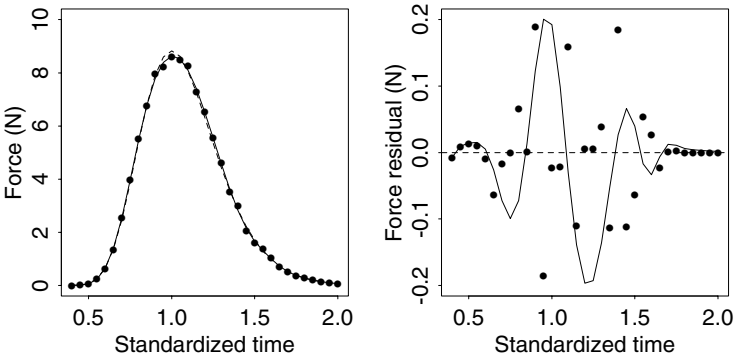

Figure 19.8. The left figure contains the data values for the first record (the points), the smoothing spline (solid curve), and the least squares fit by the model (19.8) (dotted curve). The right display shows the residuals arising from fitting the points by a spline function (the points) and the difference between the theoretical model and the spline (solid curve).

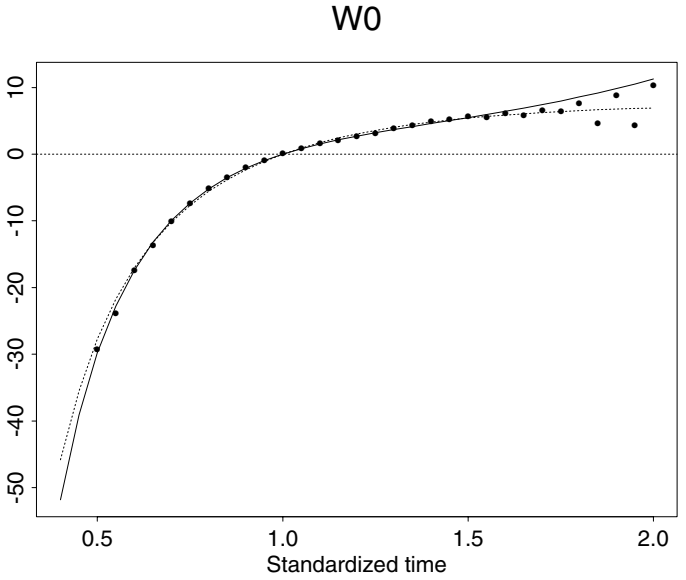

Figure 19.9. The weight function estimated by the basis expansion method for the pinch force data is indicated by the solid line, the theoretical function by the dotted line, and the point-wise estimates by the dots.

chosen after some experimentation; the first basis function was suggested by the theoretical model, and the remaining polynomial terms served to extend this model as required. Figure 19.9 shows the theoretical, the point-wise and the global estimates of the weight functions. These are admittedly close

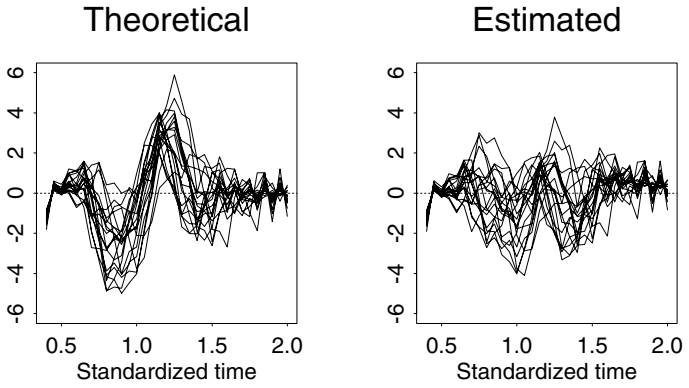

Figure 19.10. The left panel displays the forcing or impulse functions Ly_i produced by the theoretical operator, and the right panel shows the corresponding empirical operator functions.

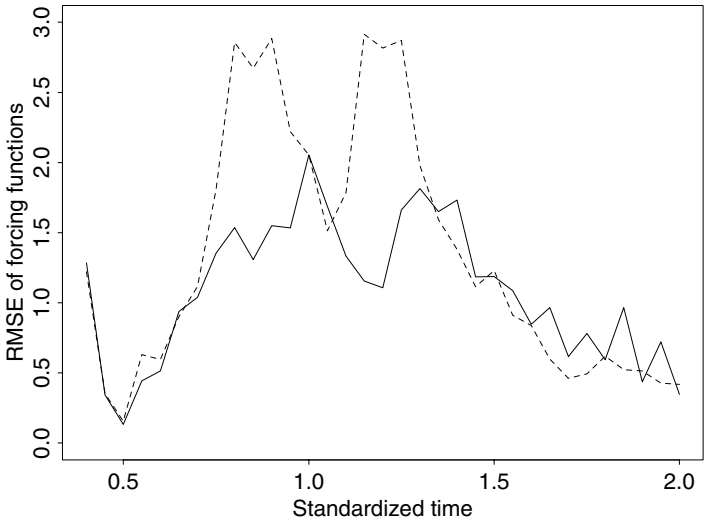

Figure 19.11. The solid line indicates the square root of the mean squared forcing function for the estimated operator, and the dotted line the same quantity for the theoretical operator.

to one another, at least in the central ranges of adjusted time, but again we observe some slight but consistent differences between the theoretical and empirical weight functions.

However, the forcing functions Ly_i, displayed in Figure 19.10, show a systematic trend for the theoretical operator, while the empirical forcing functions exhibit much less pattern. Figure 19.11 displays the root-mean-squares of the two sets of forcing functions, and this permits us to see more clearly that the estimated operator is superior in the epochs just before and after the peak force, where it produces a forcing function about half the size of its theoretical counterpart. It seems appropriate to conclude that the estimated operator has produced an important improvement in fit on either side of the time of maximum force. Ramsay, Wang and Flanagan (1995) conjecture that the discrepancy between the two forcing functions is due to drag or viscosity in the thumb-forefinger joint.

19.5 Techniques for principal differential analysis

We turn now to some methods for estimating the weight functions β_j defining the linear differential operator that comes closest to annihilating the observed functions in the sense of criterion (19.4). All but the final method assume that we have already estimated the function and its derivatives up to order m by smoothing the raw discrete data.

19.5.1 PDA by point-wise minimization

The first approach yields a point-wise estimate of the weight functions β_j computable by standard least squares estimation. Define the point-wise fitting criterion

$$\texttt{PSSE}_L(t) = \sum_i [Lx_i(t) - f_i(t)]^2 = \sum_i [\sum_{j=0}^{m} \beta_j(t) D^j x_i(t) - f_i(t)]^2, \quad (19.10)$$

where, as above, $\beta_m(t) = 1$ for all t. If t is regarded as fixed, this following argument shows that this is simply a least squares fitting criterion.

First define the m-dimensional coefficient vector

$$\boldsymbol{\beta}(t) = (\beta_0(t), \ldots, \beta_{m-1}(t))',$$

the $N \times (m+1)$ point-wise design matrix $\mathbf{Z}$ with rows

$$\mathbf{z}_i(t) = \{-x_i(t), \ldots, -D^{m-1}x_i(t), f_i(t)\}$$

and the N-dimensional dependent variable vector $\mathbf{y}$ with elements

$$y_i(t) = D^m x_i(t).$$

We can express the fitting criterion (19.10) in matrix terms as

$$\texttt{PSSE}_L(t) = [\mathbf{y}(t) - \mathbf{Z}(t)\boldsymbol{\beta}(t)]'[\mathbf{y}(t) - \mathbf{Z}(t)\boldsymbol{\beta}(t)].$$

Then, holding t fixed, the least squares solution minimizing $\mathtt{PSSE}_L(t)$ with respect to values $\beta_j(t)$ is

$$\boldsymbol{\beta}(t) = [\mathbf{Z}(t)'\mathbf{Z}(t)]^{-1}\mathbf{Z}(t)'\mathbf{y}(t). \tag{19.11}$$

The existence of these point-wise values $\boldsymbol{\beta}(t)$ depends on the determinant of $\mathbf{Z}(t)'\mathbf{Z}(t)$ being bounded away from zero for all values of t, and it is wise to compute and display this determinant as a routine part of the computation. Assuming that the determinant is nonzero is equivalent to assuming that $\mathbf{Z}(t)$ is of full column rank for all t.

Of course, if m is not large, then we can express the solution in closed form. For example, for $m = 1$ we have

$$\beta_0(t) = -\sum_i x_i(t)(Dx_i)(t) / \sum_i x_i^2(t) \tag{19.12}$$

and the full-rank condition requires that for each value of t some $x_i(t)$ be nonzero.

Some brief comments about the connections with Section 18.5.2 are in order. There, we were concerned with finding a linear operator of order m that annihilated a set of exactly m functions u_i. In order for this to be possible, an important condition was the nonsingularity of the Wronskian matrix values $\mathbf{W}(t)$ whose elements were $D^j u_i(t)$. We obtain the matrix $\mathbf{Z}(t)$ from the functions x_i in the same way, but it is no longer a square matrix, since in general we will have $N > m$. However, the condition that $\mathbf{Z}(t)$ is of full column rank is entirely analogous.

19.5.2 PDA using the concurrent functional linear model

The point-wise approach can pose problems in some applications. First, solving the equation $Lu = 0$ requires that the β_j's be available at a fine level of detail, with the required resolution depending on their smoothness. Whether or not these functions are smooth depends in turn on the smoothness of the derivatives $D^j x_i$. Since we often estimate these derivatives by smoothing procedures that may not always yield smooth estimates for higher order derivatives, the resolution we require may be very fine indeed. Moreover, for larger orders m, computing the functions β_j point-wise at a fine resolution level can be computationally intensive, since we must solve a linear equation for every value of t for which w is required. We need an approximate solution which can be quickly computed and which is reasonably regular or smooth.

It may also be desirable to circumvent the restriction that the rank of $\mathbf{Z}$ be full, especially if the failure is highly localized within the interval of integration. As a rule, an isolated singularity for $\mathbf{Z}(t)'\mathbf{Z}(t)$ corresponds to an isolated singularity in one or more weight functions β_j, and it may be desirable to bypass these by using weight functions sure to be sufficiently

smooth. More generally, we may seek weight functions more smooth or regular than those resulting from the point-wise solution.

Finally, the point-wise procedure only works if the number of functional observations N exceeds the number of columns of the point-wise design matrix $\mathbf{Z}$. But we often need to fit a differential equation to a single functional observation, and we want a method that will accommodate this case.

A strategy for identifying smooth weight functions β_j is to approximate them by using a fixed set of basis functions. This takes us directly back to Chapter 14 on the concurrent linear model, where the computational procedure is exactly what we need here. The only differences between PDA and the analyses in Chapter 14 is that here the dependent variable is $D^m x$, and the lower order derivatives can appear on the independent variable side of the equation.

19.5.3 PDA by iterating the concurrent linear model

The application of the concurrent functional linear model to this problem presupposes that the estimated derivatives $D^j x_i$ are reasonable. The melanoma analysis in Chapter 17 suggests, however, that it may be worth re-estimating the derivatives once an initial differential equation has been estimated. We can do this by using the corresponding linear differential operator to define the roughness penalty. This cycle can be repeated as many times as are required in order to achieve stable derivative estimates.

A simulated data experiment illustrates the consequences of this iterative refinement of the roughness penalty using PDA. A sample of 1000 sets of functional data were generated using the tilted sinusoid model

$$x_i(t_j) = c_{i1} + c_{i2}t_j + c_{i3}\sin(6\pi t_j) + c_{i4}\cos(6\pi t_j) + \epsilon_{ij}$$

for the 101 values $t_j = (0, 0.01, \ldots, 1)$. The coefficients $c_{ik}, k = 1, \ldots, 4$ were independently generated from a normal distribution with mean zero. The standard deviations were 1 for $k = 1$, 3 and 4, and 4 for c_{i_2}. The errors ϵ_{ij} were independent standard normal deviates. Figure 19.12 shows the first set of samples.

The errorless curves are annihilated by the operator

$$Lx = (6\pi)^2 D^2 x + D^4 x,$$

where $(6\pi)^2 = 355.3$. How well can we estimate this operator from these data? Does estimating this operator buy us anything in terms of the qualities of the estimates of the curves and their derivatives? For example, how well is the second derivative estimated when we use an estimated operator L rather than the default choice $L = D^4$?

The initial operator was consequently $L = D^4$, and was used to define the initial penalty matrix $\mathbf{R}$. The basis system that we used to estimate the true curves and their derivatives consisted of 105 order 6 B-spline basis

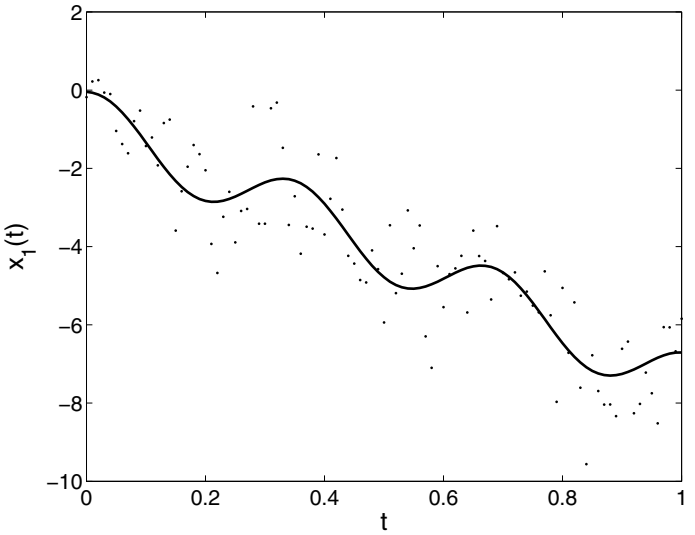

Figure 19.12. The dots indicate the data generated by adding independent standard normal deviates to the tilted sinusoid shown as the solid line.

functions with knots at the sampling points t_j. We cycled through the following process five times:

1. The value of the smoothing parameter λ minimizing the GCV criterion was found using a numerical optimization method. In order to avoid rounding error problems, an upper limit on the allowable estimate was set to $10 - \log_{10} \text{trace}\mathbf{R}$.

2. The data were smoothed using this value of λ.

3. A principal differential analysis was performed based on the concurrent linear model method described above. All four coefficient functions $\beta_j, j = 0, 1, 2, 3$ were estimated using the constant basis for each.

4. The linear differential operator estimated by PDA was then used to redefine the penalty matrix $\mathbf{R}$.

The estimated λ after the first cycle was $10^{-9.9}$, and after the second cycle it came up against the upper limit that we imposed, which for these data was $10^{-8.8}$. Subsequent iterations hardly changed the results at all.

After the first iteration, defined by $L = D^4$, the PDA estimated the operator as

$$Lx = 2360.3x - 123.8Dx + 376.1D^2x - 0.3D^3x + D^4x$$

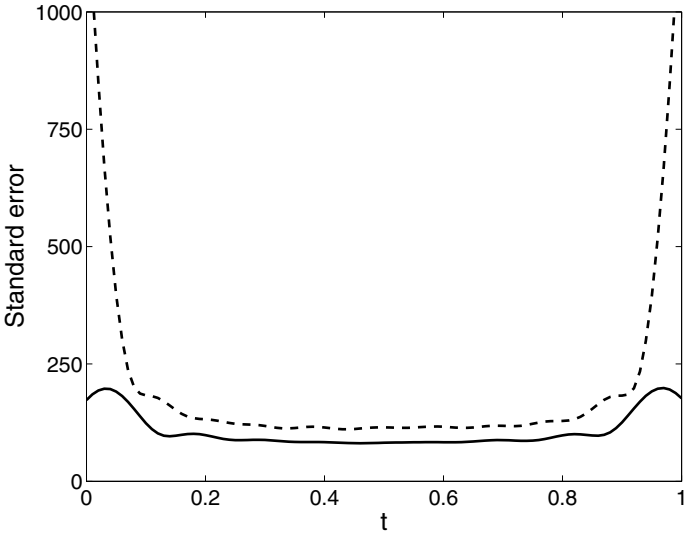

Figure 19.13. The solid curve is the standard error of estimate for the second derivative of the tilted sine data after five iterations, and the dashed line is after the first iteration.

and after all five iterations, the estimate was

$$Lx = 310.4x - 125.4Dx + 357.0D^2x - 0.4D^3x + D^4x.$$

The estimate after one iteration is quite good, judging by the coefficients for the key derivatives of order 2 and 3 that determine the period and phase of the sinusoid, and only slightly improved by going through all five iterations. For the record, we also tried fixing the first two coefficients to zero, but the estimates of the second two coefficients were not appreciably better.

The most striking benefit is in terms of the precision of the function and derivative estimates. The ratios of the first iteration integrated mean squared error to that on the fifth iteration are 1.2, 2.0, 3.8, 5.3 and 8.1 for derivatives of order 0, 1, 2, 3 and 4, respectively. The function values are modestly improved, but the improvement brought about by iterative refinement of L increases with the order of the derivative. To see better both the improvement and how it is achieved, we turn to Figure 19.13 which shows the point-wise standard errors of the second derivatives after the first and fifth iterations. The big impact is at the endpoints, where estimating a linkage between the function value and the second derivative greatly diminishes the standard error. Because function value estimates are less affected by having half the number of neighbors at the endpoints than are derivatives, estimating this linkage passes along the function value stability to the derivatives.

These results are probably better than we would encounter in practice, mainly because the true curves could all be annihilated in principle by a linear differential operator within the family of those that we could estimate. A similar study of growth curves generated by the Jolicoeur model used in Chapters 5 and 6 came up with a much more modest improvement in the second derivative because the variation from curve to curve is more complex than can be modelled with a single order four operator. Nevertheless, the improvements there were also more pronounced at the endpoints.

19.5.4 Assessing fit in PDA

Since the objective of PDA is to minimize the norm $\|L\mathbf{y}\|$ of the forcing function associated with an estimated differential operator, and since the quality of fit can vary over the domain T, it seems appropriate to assess fit in terms of the point-wise error sum of squares $\texttt{PSSE}_L(t)$ as defined in (19.10). As in linear modelling, the logical baseline against which we should compare $\texttt{PSSE}_L$ is the error sum of squares defined by a theoretical model and its associated weight functions ω_j:

$$\texttt{PSSE}_0(t) = \sum_i [\sum_{j=0}^{m-1} \omega_j(t)(D^j y_i)(t) + (D^m y_i)(t)]^2. \tag{19.13}$$

In the event that there is no theoretical model at hand, we may use $\omega_j = 0$, so that the comparison is simply with the sum of squares of the $D^m y_i$. From these loss functions, we may examine the point-wise squared multiple correlation function

$$\texttt{RSQ}(t) = \frac{\texttt{PSSE}_0(t) - \texttt{PSSE}_L(t)}{\texttt{PSSE}_0(t)} \tag{19.14}$$

and the point-wise F-ratio

$$\texttt{FRATIO}(t) = \frac{(\texttt{PSSE}_0(t) - \texttt{PSSE}_L(t))/m}{\texttt{PSSE}_0(t)/(N-m)}. \tag{19.15}$$

19.6 Comparing PDA and PCA

19.6.1 PDA and PCA both minimize sums of squared errors

Once we have found the operator L, we can in general define m linearly independent functions $\xi_1, \ldots, \xi_m$ that span the null space of L, so that any function x that satisfies $Lx = 0$ can be expressed precisely as a linear combination of the ξ_j. This means that the functions ξ_j form a basis for this space of solutions. Just how we compute such a basis is taken up in Chapter 18.

Let us assume that we have a sample of N observed functions x_i, where $N = 1$ is allowed. If these functions are not necessarily solutions to (19.3), then we can quantify the extent to which they approach being solutions by the size of the forcing functions ϵ_i defined by

$$Lx_i = \epsilon_i.$$

An example of this idea was given in Figure 1.7 where we applied the harmonic acceleration operator to four temperature profiles and discovered that these forcing functions were substantially nonzero.

The algorithm that we used in Section 17.3 aimed, at each iteration, to find the operator L that minimized the integrated square of the residual function ϵ. Why? Because it used the concurrent functional linear model developed in Chapter 14 to minimize a measure of discrepancy between the derivative that acted as the dependent variable and the fit based on two lower order derivatives that acted as independent variables. In effect, this minimizes a sum of squares measure for the corresponding L operator.

Consequently, we have a technique for choosing L so as to make the Lx_i as small as possible. If the technique is successful, then the residual functions will be small relative to the highest order of derivative. We should then expect to obtain a good approximation of the x_i by expanding them in terms of the ξ_j that span the subspace defined by the corresponding differential equation.

This is closely reminiscent of PCA, where the first m principal component functions ξ_j also define an m-dimensional subspace for approximating the given data.

19.6.2 PDA and PCA both involve finding linear operators

We can pursue the comparison between PCA and PDA further by noting that PCA can also be considered to involve the identification of a linear operator, which we can denote by Q, such that the equation $Qx_i = 0$ is solved as nearly as possible. To see this, recall from Chapter 8 that the goal of functional PCA is to find a set of m basis functions ξ_j such that the least squares criterion

$$\texttt{SSE}_{\text{PCA}} = \sum_{i=1}^{N} \int [x_i(t) - \sum_{j=1}^{m} f_{ij}\xi_j(t)]^2 \, dt \tag{19.16}$$

is minimized with respect both to the basis functions ξ_j and with respect to the coefficients of the expansions of each observed principal component score f_{ij}.

Because the fitting criterion (19.16) is least squares, we can think of PCA as a two-stage process: First identify a set of m orthonormal basis functions ξ_j, and then approximate any specific curve x_i by $\hat{x}_i = \sum_{j=1}^{m} f_{ij}\xi_j$. This second basis expansion step is the projection of each of the observed

functions onto the m-dimensional space spanned by the basis functions ξ_j, and takes place after having first identified the optimal basis for these expansions. Thus $\hat{x}_i$ is the image of x_i resulting from applying a least squares fit.

Suppose we indicate this projection as P_ξ, with the subscript indicating that the nature of the projection depends on the basis functions ξ_j. That is, $P_\xi x_i = \hat{x}_i$.

Associated with the projection P_ξ is the complementary projection

$$Q_\xi = I - P_\xi,$$

which produces as its result the residuals

$$Q_\xi x_i = x_i - P_\xi x_i = x_i - \hat{x}_i.$$

Using these projection operators, we can alternatively and equivalently define the PCA problem in a way that is much more analogous to the problem of identifying the linear differential operator L: In PCA, one seeks a projection operator Q_ξ such that the residual sum of squares

$$\mathtt{SSE}_{\mathrm{PCA}} = \sum_{i=1}^{N} \int [Q_\xi x_i(t)]^2 \, dt \tag{19.17}$$

is minimized. Indeed, one might think of the first m eigenfunctions as the functional *parameters* defining the projection operator Q_ξ, just as the weight functions β are the functional parameters defining L in PDA. These eigenfunctions, and any linear combinations of them, exactly satisfy the equation $Q_\xi \xi_j = 0$, just as the m functions ξ_j referred to above exactly satisfy the equation $L_\beta \xi_j = 0$, where we now add the subscript β to L to remind ourselves that L is defined by the vector $\boldsymbol{\beta}$ containing the m weight functions β_j.

Principal differential analysis is defined, therefore, as the identification of the differential operator L_β that minimizes least squares criterion $\mathtt{SSE}_{\mathrm{PDA}}$; principal components analysis is defined as the identification of the projection operator Q_ξ that minimizes the least squares criterion $\mathtt{SSE}_{\mathrm{PCA}}$. Both operators are linear.

19.6.3 Differences between differential operators (PDA) and projection operators (PCA)

Since the basic structures of the least squares criteria (19.17) and (19.4) are the same, clearly the only difference between the two criteria is in terms of the actions represented by the two operators L_β and Q_ξ. Since $Q_\xi x$ is in the same vector space as x, the definition of the operator identification problem as the minimization of $\|Q_\xi x\|^2$ is also in the same space, in the sense that we measure the performance of Q_ξ in the same space as the functions x to which it is applied.

On the other hand, L_β is a roughening transform in the sense that $L_\beta x$ has m fewer derivatives than x and is usually more variable. We may want to either penalize or otherwise manipulate x at this rough level.

Put another way, it may be plausible to conjecture that the noise or unwanted variational component in x is found only at the rough level $L_\beta x$. Thus, a second motivating factor for the use of L_β rather than Q_ξ is that PDA process explicitly takes account of the smoothness of the data by first roughening the data before minimizing error, while PCA does not.

Once we have found the operator L, we can in general define m linearly independent functions $u_1, \ldots, u_m$ that span the null space of L, and so any function x that satisfies $Lx = 0$ can be expressed precisely as a linear combination of the u_j. Hence, since L has been chosen to make the Lx_i as small as possible, we would expect to obtain a good approximation of the x_i by expanding them in terms of the u_j. This is closely reminiscent of PCA, where the first m principal component functions ξ_j form a good m-dimensional set for approximating the given data. The spirit of the approximation is rather different, however.

We can pursue the comparison between PCA and PDA by noting that PCA can also be considered to involve the identification of a linear operator, which we can denote by Q, such that the equation $Qx_i = 0$ is solved as nearly as possible. To see this, recall from Chapter 8 that one method of defining functional PCA is to propose to find a set of m basis functions ξ_j such that the least squares criterion

$$\mathtt{SSE}_{\mathrm{PCA}} = \sum_{i=1}^{N} \int [x_i(t) - \sum_{j=1}^{m} f_{ij}\xi_j(t)]^2 \, dt \tag{19.18}$$

with respect both to the basis functions ξ_j and with respect to the coefficients of the expansions of each observed function, f_{ij}. Because the fitting criterion (19.18) is least squares, we can think of PCA as a two-stage process: first identify a set of m orthonormal basis functions ξ_j, and then approximate any specific curve x_i by $\hat{x}_i = \sum_{j=1}^{m} f_{ij}\xi_j$. This second basis expansion step is the projection of each of the observed functions onto the m-dimensional space spanned by the basis functions ξ, and takes place after having first identified the optimal basis for these expansions. Thus $\hat{x}_i$ is the image of x_i resulting from applying a least squares fit.

Suppose we indicate this projection as P_ξ, with the subscript indicating that the nature of the projection depends on the basis functions ξ_j. That is, $P_\xi x_i = \hat{x}_i$. Associated with the projection P_ξ is the complementary projection

$$Q_\xi = I - P_\xi,$$

which produces as its result the residuals

$$Q_\xi x_i = x_i - P_\xi x_i = x_i - \hat{x}_i.$$

Using this concept, we can alternatively and equivalently define the PCA problem in a way that is much more analogous to the problem of identifying the linear differential operator L: In PCA, one seeks a projection operator Q_ξ such that the residual sum of squares

$$\mathtt{SSE}_{\mathrm{PCA}} = \sum_{i=1}^{N} \int [Q_\xi x_i(t)]^2 \, dt \tag{19.19}$$

is minimized. Indeed, one might think of the first m eigenfunctions as the functional *parameters* defining the projection operator Q_ξ, just as the weight functions w are the functional parameters of the LDO L in PDA. These eigenfunctions, and any linear combinations of them, exactly satisfy the equation $Q_\xi \xi_j = 0$, just as the m functions u_j referred to above exactly satisfy the equation $L_w u_j = 0$, where we now add the subscript w to L to remind ourselves that L is defined by the vector w of m weight functions β_j.

Principal differential analysis is defined, therefore, as the identification of the operator L_w that minimizes least squares criterion $\mathtt{SSE}_{\mathrm{PDA}}$, just as we can define PCA as the identification the projection operator Q_ξ that minimizes the least squares criterion $\mathtt{SSE}_{\mathrm{PCA}}$.

Since the basic structures of the least squares criteria (19.19) and (19.4) are the same, clearly the only difference between the two criteria is in terms of the actions represented by the two operators L_w and Q_ξ. Since $Q_\xi x$ is in the same vector space as x, the definition of the operator identification problem as the minimization of $\|Q_\xi x\|^2$ is also in the same space, in the sense that we measure the performance of Q_ξ in the same space as the functions x to which it is applied.

On the other hand, L_w is a roughening transform in the sense that $L_w x$ has m fewer derivatives than x and is usually more variable. We may want to either penalize or otherwise manipulate x at this rough level. Put another way, it may be plausible to conjecture that the noise or unwanted variational component in x is found only at the rough level $L_w x$. Thus, a second motivating factor for the use of L_w rather than Q_ξ is that PDA process explicitly takes account of the smoothness of the data by first roughening the data before minimizing error, while PCA does not.

As an example, imagine that we are analyzing the trajectories x_i of several rockets of the same type launched successively from some site. We observe that not all trajectories are identical, and we conjecture that some random process is at work that contributes variability to our observations. Naively, we might look for that variability in the trajectories themselves, but our friends in physics will be quick to point out that, first, the major source of variability is probably in the propulsion system, and second since the force that it applies is proportional to acceleration, we ought to study the acceleration $D^2 x_i$ instead. That is, if the function x_i is the trajectory along a specific coordinate axis (straight up, for example), the systematic

or errorless trajectory should obey the law

$$f_i(t) = M(t)D^2x_i(t),$$

where $M(t)$ is the mass of the rocket at time t. Alternatively,

$$-f_i/M + D^2x_i = 0.$$

Taking a more empirical approach, however, we agree on the compromise of looking for a second order linear differential equation

$$Lx = \beta_0 x + \beta_1 Dx + D^2x$$

and, if our friends in physics are right, the systematic or errorless component in the data should yield

$$\beta_0 x_i = -f_i/M \text{ and } \beta_1 = 0.$$

What we do understand, in any case, is that the sources of variability are likely to be at the rough level D^2x_i, rather than at the raw trajectory level x_i.

Returning to the lip position curves, we might reason that variation in lip position from curve to curve is due to variation in the forces resulting from muscle contraction, and that these forces have a direct or proportional impact on the acceleration of the lip tissue, and thus only indirectly on position itself. Position is two derivatives away from the action, in short.

More generally, an important motivation for finding the operator L_w is substantive: Applications in the physical sciences, engineering, biology and elsewhere often make extensive use of differential equation models of the form

$$Lx_i = f_i.$$

The result f_i is often called a *forcing or impulse function*, and in physical science and engineering applications is often taken to indicate the influence of exogenous agents on the system defined by $Lx = 0$.

Section 19.5 presents techniques for principal differential analysis, along with some measures of fit to the data. We also take up the possibility of regularizing or smoothing the estimated weight functions β_j.

19.7 Further readings and notes

Viele (2001) also analyzed the pinch force data with an alternative strategy for testing whether the model (19.8) adequately fits the data.