# Representing Penalties on Basis Expansions as a Polynomial In the Parameters (Working Title)

March 8, 2013

## 1 Basis Function Expansion Methods

Fitting sophisticated mathematical functions to empirical data continues to be a challenge in statistical science. One approach, known as the basis function expansion method offers a considerable degree of flexibility and mathematically tractable solutions. This approach involves representing an unknown function $y(t)$ using a finite number of basis functions $\{\phi_k(t)\}$ through the conditional expectation

$$\mathbb{E}[y|t] = c_0\phi_0(t) + c_1\phi_1(t) + \cdots + c_N\phi_N(t),$$

where $\{c_k\}$ is a set of appropriately chosen coefficients.

Here, $\mathbb{E}[y|t]$ is the conditional expectation of the dependent variable $y$, given our independent data $x$. The conditional expectation $\mathbb{E}[y|t]$ is the estimate of $y$ depending exclusively on $t$ that minimises the squared loss $\mathbb{E}[y(t) - E[y|t]]^2$. This is a theoretical optimiser, and so in practice we must make do with an estimate $\widehat{\mathbb{E}[y|t]}$ instead.

**Example 1.** (Simple Linear Regression) Simple linear regression is an example of a statistical procedure that uses a basis function representation. Take two basis functions $\{1, t\}$, so that $\hat{y}(t)$, our estimate of $y$ given the value $t$, can be written as a linear combination of the two $\widehat{\mathbb{E}[y|t]} = \hat{y}(t) = c_0 + c_1 t$.

This is the form of a simple linear regression model. If our basis consists of just the single constant function $\phi(t) = 1$ then we get a model of the form $\hat{y} = c_0$.

In this case we would generally use the mean of the $y$ values as our estimate of $c_0$. If we go in the other direction and add a quadratic function $t^2$ we get a quadratic regression model $\hat{y}(t) = c_0 + c_1 t + c_2 t^2$.

□

**Example 2.** (Broken Stick Function) Another choice of basis is the two functions $\{\phi_1, \phi_2\}$ defined as follows

$$\begin{aligned}\phi_1(t) &= 1 \\ \phi_2(t) &= \begin{cases} 0 & \text{if } t < \tau \\ t - \tau & \text{if } t \geq \tau. \end{cases}\end{aligned}$$

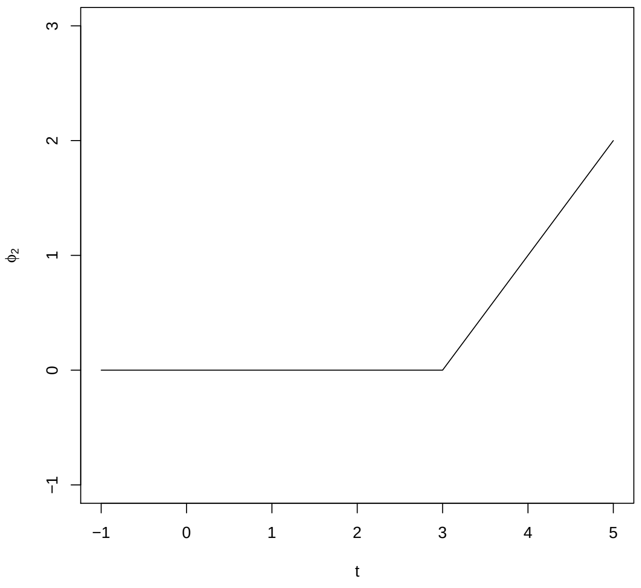

Figure 1: The broken stick function $\phi_2$.

The first function is a constant function, the second is a ramp function based at $\tau$.

This basis would work very well for representing data which starts to increase linearly past a certain threshold. Broken stick functions and constant functions are very cheap computationally, only needing literally a line or two of code to implement and only have at most one associated parameter ($\tau$), consuming little memory. A computer can produce and work with comparatively a very large number of them at once. This could be used to cover situations where there data is unchanging over relatively long periods.

Ramp functions are the building blocks for the "p-splines" developed by Eiler and Marx. P-splines are a precursor to the penalty based approach used here. P-splines penalise the coefficients, whereas we will penalise the generated curve.

The above basis has two disadvantages though. Firstly the second basis function is only continuous, not differentiable. It would not be wise to use this basis if we wanted to estimate d$y$/d$t$. Secondly it is very arbitrary. It is not obvious why it would be useful compared to a quadratic model for example; the latter also allows to us estimate derivatives of all orders and can still accommodate data which varies in its rate of change.

□

This underscores an important point. There are many types of basis besides the ones usually encountered, such as polynomial bases. Many of them are fungible from the pure mathematical point of view in terms of how well they can fit a function. The statistician should make a sensible choice of basis. If our choice of basis is good, then it will be able to fit the data with only a few terms, and we might be able to avoid estimating many coefficients. This is one of the reasons why simple linear regression and quadratic regression are useful; they can capture much variation in the observed data in spite of being very simple.

**Example 3.** (Fourier Basis Functions) If our data has a periodic component to it, such as the observed temperature over the course of the year, or a time series of annual sales, then it seems intuitively correct to use a basis consisting of periodic functions. This suggests that we should use a Fourier basis consisting of the set of functions $\{\cos(n\omega t), \sin(n\omega t)\}$ where $0 \leq n \leq N$ for some $N$ and $\omega$ is the frequency. The frequency depends on our time scale, they are related by the formula $\omega = 2\pi/T$, where $T$ is the period.

Furthermore Fourier basis functions have several other desirable properties. They are smooth, meaning that they can be used to estimate any derivative of the data, at least in theory. They are orthogonal, which can make certain problems more convenient and they are closed under differentiation, meaning that the derivative of a combination of Fourier basis functions, is itself a combination of Fourier basis functions. The latter two properties will prove very useful later, as we shall see.

A Fourier basis can represent any square integrable function on some interval. This covers a large proportion of the functions encountered in real life They cannot detect how frequencies change in space however, only their global behaviour. They can also be somewhat mundane - almost certain to work, but unlikely to provide anything too interesting.

□

**Example 4.** (B-spline Basis Functions) Roughly speaking, B-splines are compactly supported polynomial functions, or more practically they are nonzero only inside of a given interval. More formally a B-Spline basis consists of a degree $n$, which determines the degree of the basis functions, and a set of knots, that is a set of $K$ time points $\{t_0, \ldots, t_K\}$.

Since they are compact they generally can only individually capture local information about the data. One of the main advantages of B Splines is that they can represent any other spline of the same degree and smoothness with the same knots. This makes them useful for statistics since make fewer assumptions less about the form of the data, they can help us avoid bias.

□

# 2 Least Squares Fitting

Least squares fitting is a means of fitting a function to data. The least squares fit is the one that minimises the sums of the squared errors. In the cases where we have $n$ observations $y_i$ measured at times $t_i$ and where we are selecting our estimated function $\hat{y}(t)$ from a set of functions $S$, the least squares fit $\hat{y}_{LS}(t)$ is defined as

$$\hat{y}_{LS}(t) = \operatorname*{argmin}_{f \in S} \sum_{i=1}^{n} [y_i - f(t_i)]^2.$$

From now on we will always be working with the sums of the squared errors, so we will not bother using any subscripts to indicate this and only use $\hat{y}(t)$ rather than $\hat{y}_{LS}(t)$.

Assume $S$ can be spanned by some set of $m$ basis functions i.e. $S = \{\sum_{i=1}^{m} c_i\phi_i(t) | c_i \in \mathbb{R}\}$. This suggests that to find $\hat{y}(t)$ it is only necessary to estimate to coefficients $\hat{c}_i$, so that $\hat{y}(t)$ never appears explicitly. Then we have completely determined $\hat{y}$. We can then write the least squares criterion as:

$$(\hat{c}_1, \ldots, \hat{c}_m) = \underset{c_1 \ldots c_K}{\operatorname{argmin}} \sum_{i=1}^{n} [y_i - \sum_{j=1}^{m} c_j\phi_j(t_i)]^2$$

The above expression is can be expressed more cleanly using vector notation. Firstly, we have $f(t) = \mathbf{c}'\phi(\mathbf{t})$, where $\mathbf{c} = (c_1, ..., c_m)$ and $\phi = (\phi_1(t), \ldots, \phi_2(t))'$. By constructing a matrix $\mathbf{\Phi}$, where the $i$th row of $\mathbf{\Phi}$ is $\phi(t_i)$, and let $\mathbf{y} = (y_1, \ldots, y_n)$ the least squares problem can written as

$$\hat{\mathbf{c}} = \underset{\mathbf{c} \in \mathbb{R}^m}{\operatorname{argmin}} (\mathbf{y} - \mathbf{\Phi c})'(\mathbf{y} - \mathbf{\Phi c}).$$

The expression on the right hand side is an example of a quadratic form, they are the generalisation of quadratic functions to finite dimensional vector spaces. We can have quadratic forms on infinite dimensional vectors spaces too, but that is not relevant here.

The advantage of quadratic forms is that they are very easy to minimise as any textbook on regression or numerical optimisation will tell you.

# 3 Roughness Penalties

It is well known that the least squares gives us the BLUE of $\mathbf{y}$, the function we assume to be generating the data. Nonetheless it is often useful to employ a form of regularisation which constrains how much $\hat{y}$ is allowed to vary. Intuitively, this reduces the variation of $\hat{y}$ and helps guard against overfitting. In the context of basis function expansions, the most commonly used penalty is the curvature penalty

$$\text{PEN}(\hat{y}) = \int_D [\hat{y}''(t)]^2 \mathrm{d}t.$$

Here $D$ is our domain of interest and $\hat{y}''(t)$ stands for the second derivative of the function $\hat{y}(t)$.

This penalty tries to force $\hat{y}(t)$ towards solutions of the differential equation $\hat{y}''(t) = 0$. The general solution of this ODE is of the form $\hat{y}(t) = \alpha + \beta t$, a straight line. The penalty nudges $\hat{y}(t)$ towards linear regression.

This penalty can also be represented as a polynomial. Let $\langle f, g \rangle = \int_D f(t)g(t)\mathrm{d}t$. Here, $\langle \cdot, \cdot \rangle$ is the inner product on $L^2(D)$, the set of square integrable functions on $D$. The penalty can be written in the form $\text{PEN}(f) = \langle f'', f'' \rangle$. Substituting in the expansion for $f$ we get

$$
\begin{aligned}
\langle f'', f''\rangle &= \langle \sum_{i=1}^{m} c_i\phi_i'', \sum_{i=1}^{m} c_i\phi_i''\rangle \\
&= \sum_{i=1}^{m}\sum_{j=1}^{m}\langle c_i\phi_i'', c_j\phi_j''\rangle \\
&= \sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j\langle \phi_i'', \phi_j''\rangle.
\end{aligned}
$$

Here, $\langle \phi_i'', \phi_j''\rangle$ depends only on our choice of basis and so are "fixed" for our purposes. This implies that the penalty can be represented as a polynomial in the $c_i$. We can do better however and represent the penalty as a quadratic form. If we define an $m \times m$ matrix $\mathbf{K}$ by $\mathbf{K}_{ij} = \langle \phi_i'', \phi_j''\rangle$ it can be seen that $\langle f'', f''\rangle = \mathbf{c}'\mathbf{K}\mathbf{c}$.

Note that this relies on the fact that the second derivative operator maps a linear combination of $\phi_i$ into a linear combination $\phi_i''$. We can always represent an inner product on some finite vector space as $\mathbf{b}'\mathbf{A}\mathbf{b}$, where the terms depend on the problem at hand. We can simply construct the matrix $\mathbf{A}$ for the $\phi_i''$ and have the $\mathbf{c}$ in place of the $\mathbf{b}$ because differentiation is linear.

# 4 Penalised Least Squares

There are two distinct, but competing objectives. We want a fit $\hat{y}$ that maintains fidelity to the data, as represented by the goodness of fit, but simultaneously adheres to the requirement of a smooth description of the data.

The solution is to use a penalty that incorporates both of these objectives, the penalised sum of squared errors. The penalised sum of squared errors is a functional, or a function whose domain is a set of functions and whose codomain is the real numbers. Here it is a functional on the spanning set of our basis functions $\{\phi_i(t)\}$. It is the weighted sum of the roughness and least squares penalties

$$\mathrm{PENSSE}(f) = \sum_{i=0}^{n}(y_i - f(t_i)^2 + \lambda \int_D [f''(t)]^2 \mathrm{d}t.$$

The parameter $\lambda$ controls the tradeoff between error and smoothness as represented by the least squares and roughness terms respectively. It can be thought of as a model complexity parameter; as $\lambda$ increases $\hat{y}(t)$ becomes more linear and so our model tends more towards simple linear regression, as it decreases the model tends more towards fitting the data exactly.

In polynomial regression the analogous parameter would be the order of the polynomial we are fitting. However the two quantities differ in that the order of a polynomial is a discrete quantity, whilst $\lambda$ is a continuous one.

We can also think of $\lambda$ as controlling the degree of fidelity of the fitted curve to the ODE $\hat{y}''(t) = 0$.

The choice of $\lambda$ is important; strictly speaking we ought to denote $\hat{y}(t)$ as $\hat{y}_\lambda(t)$ to denote the dependence. An estimate for $\lambda$ can generally be found by cross validation. As discussed in Green and Silverman sometimes there are better choices of $\lambda$ than a computer can find though.

Assume again that $\hat{y}(t)$ is the sum of basis functions, depending on a vector of coefficients $\hat{\mathbf{c}}$. This means we can omit the dependence on $\hat{y}(t)$ and convert from the problem of estimating a function to estimating a vector $\hat{\mathbf{c}}(\lambda)$ (or just simply $\hat{\mathbf{c}}$) which depends on our choice of smoothing parameter. Using the previous results we can show that the Penalised Sum of Squares can be written as the sum of two quadratic forms:

$$\text{PENSSE}(c) = (\mathbf{y} - \mathbf{\Phi c})'(\mathbf{y} - \mathbf{\Phi c}) + \lambda \mathbf{c}'\mathbf{K}\mathbf{c}.$$

Where $\mathbf{y}$, $\mathbf{K}$ and $\mathbf{\Phi}$ have the same definitions as they did before. This is an regularised regression problem or Tikhonov regularisation. The problem of minimising this expression has the solution

$$\begin{aligned}\hat{\mathbf{c}}(\lambda) &= (\mathbf{\Phi}'\mathbf{\Phi} + \lambda\mathbf{K})^{-\mathbf{1}}\mathbf{\Phi}'\mathbf{y} \\ &= \mathbf{H}(\lambda)\mathbf{y}\end{aligned}$$

If this problem is looked at from the perspective of linear models, as in Christessen, then $\mathbf{y} = \mathbf{\Phi c} + \mathbf{e}$, where the elements of $\mathbf{e}$ are i.i.d and have mean zero and variance $\sigma^2$, then we can estimate the variance of $\hat{y}(t)$. We will denote the hat matrix $\mathbf{H}(\lambda)$ by $\mathbf{H}$.

We have $\hat{\mathbf{c}} = \mathbf{H}(\mathbf{\Phi c} + \mathbf{e})$, so $\mathbb{E}[\hat{\mathbf{c}}] = \mathbf{H\Phi c}$. Note $\hat{\mathbf{c}}$ can be a biased estimate of $\mathbf{c}$. The variance of $\hat{\mathbf{c}}$ is given by $\text{Var}[\hat{\mathbf{c}}] = \sigma^2\mathbf{H\Phi\Phi'H'}$. Hence the expected value of $\hat{\mathbf{y}}$ is $\mathbf{\Phi H\Phi c}$ and its variance is $\sigma^2\mathbf{\Phi H\Phi\Phi'H'\Phi'}$.

# 5 Multivariate Bases

What if our data is spatial in nature? In this case our data will often be at a series of points $\mathbf{x}_i = (x_i, y_i); i = 1, \ldots, n$. It is actually not too difficult to extend our results; it is almost as easy as replacing the $t_i$ with $\mathbf{x_i}$.

As before we expand our function $y$ as a basis function expansion

$$y(\mathbf{x}) = \sum_{i=1}^{m} c_i\phi(\mathbf{x}).$$

Notice that even though we are working in more than one dimension, our sum is still "one dimensional" in that it has only one index. This is deliberate as it makes our life much easier.

Finding a least squares estimate is identical, so we will not cover it here.

**Example 5.** Fitting the Displacement of a Membrane Using a Laplacian Penalty

On model of the mechanics of a membrane is the wave equation. Let $u(x, y, t)$ be the displacement of the membrane at position $(x, y)$ at time $t$. Then $u$ is assumed to satisfy the partial differential equation:

$$u_{tt} = c\Delta u.$$

Here $u_{tt} = \frac{\partial^2 u}{\partial t^2}$, $c$ is a parameter known as the wave speed and $\Delta$ is a differential operator known as the Laplacian. In two dimensions it is defined as

$$\Delta = \frac{\partial^2}{\partial x^2} + \frac{\partial^2}{\partial y^2}.$$

If the membrane doesn't move with time so $u_{tt} = 0$ then the wave equation reduces to Laplace's equation

$$\Delta u = 0.$$

This is what is known as a steady state model, notice the coefficient $c$ has disappeared.

If we have measurements of the displacements at various points, this suggests that we use a penalised model of the form

$$PENSSE(f) = \sum_{i=0}^{n} (y_i - f(\mathbf{x}_i))^2 + \lambda \int_\Omega |\Delta f(\mathbf{x})|^2 \mathbf{dx}.$$

As before we can express this penalty as a quadratic form. Define an inner product on our functions by

$$\langle f, g\rangle = \int_\Omega f(\mathbf{x})g(\mathbf{x})\mathbf{dx}.$$

Then the laplacian penalty can be written as

$$\int_\Omega |\Delta f(\mathbf{x})|^2 \mathbf{dx} = \langle \Delta f, \Delta f\rangle.$$

If $f = \sum c_i\phi_i$ then $\Delta f = \sum c_i \Delta\phi_i$. In the same manner as the previous roughness penalty, we can see that if f is a basis expansion then we can write the penalty as a polynomial in the coefficients

$$\langle \Delta f, \Delta f\rangle = \sum_{i=1}^{m}\sum_{j=1}^{m} c_i c_j \langle \Delta\phi_i, \Delta\phi_j\rangle.$$

This can of course be written as $\mathbf{c}'\mathbf{K}\mathbf{c}$ where $\mathbf{K}_{ij} = \langle \Delta\phi_i, \Delta\phi_j\rangle$.

We define the matrix $\mathbf{\Phi}$ as before with $\mathbf{\Phi}_{ij} = \phi_j(\mathbf{x}_i)$, $\mathbf{c} = (c_1, \ldots, c_m)$. We can then write the penalty in terms of the coefficients

$$PENSSE(c) = (\mathbf{y} - \mathbf{\Phi}\mathbf{c})'(\mathbf{y} - \mathbf{\Phi}\mathbf{c}) + \lambda\mathbf{c}'\mathbf{K}\mathbf{c}.$$

It is worth noting that this is identical to the expression we derived above. This is because both $\frac{d^2}{dt^2}$ and $\Delta$ are linear operators.

# 6 Non Linear Penalties

Suppose we believed that our data could be approximately modelled by and ODE of the form $f'' = f^2$ and we wished to incorporate this data into our model. It would be reasonable to include a penalty of the form:

$$\int_D (f''(t) - f(t)^2)^2 \mathrm{dt}$$

We can also represent this penalty as a quadratic form. As usual $f = \sum c_i \phi_i$. Expanding out the two terms in the penalty we get

$$f''(t) = \sum_{i=0}^{m} c_i \phi_i(t)''$$

$$f(t)^2 = \sum_{i=0}^{m} \sum_{j=0}^{m} c_i c_j \phi_i(t) \phi_j(t).$$

Hence,

$$f''(t) - f(t)^2 = \sum_{i=0}^{m} c_i \phi(t)'' - \sum_{i=0}^{m} \sum_{j=0}^{m} c_i c_j \phi_i(t) \phi_j(t).$$

Notice the second term is a two dimensional, finite sum. We now need to find norm. As before we can write this as an inner product: $\langle f'' - f^2, f'' - f^2 \rangle$. The above expression is in the form $a_1 \phi_1'' + \cdots + a_m \phi_m'' + b_{11} \phi_1 \phi_1 + \ldots b_{mm} \phi_m \phi_m$. This is a linear combination of a finite set of functions. So we can represent it as a quadratic form. The form above is a little awkward to work with. We would like to have it vary with one index only, or combine the $\phi_i'''$ and the $\phi_i \phi_j$ together. We would define a function $\pi(n)$ that returns the appropriate function either some $\phi_i$ or $\phi_i \phi_j$ depending on the value. Defining such a function is tricky however. One function is as follows

$$\pi(k) = \begin{cases} \phi_k'' & \text{if } k \leq m \\ \phi_{(k-1)|m} \phi_{(k-1) \bmod m} & \text{if } k > m \end{cases}$$

Here $a|b$ is integer division, i.e. $a|b = \text{floor}(a/b)$.

If we define $\psi_k(t)$ to be $\pi_k(t)$ and define a similar function $\sigma(i)$ for the coefficients we can define $f'' - f^2$ we get

$$f'' - f^2 = \sum \sigma(i) \psi_i.$$

We now can express the penalty as a quadratic form

$$\int_D (f(t)'' - f(t)^2)^2 dt = \sigma(\mathbf{c})' \mathbf{K} \sigma(\mathbf{c}).$$

Here $\mathbf{K}_{ij} = \langle \psi_i, \psi_j \rangle$ and $\sigma(\mathbf{c}) = (\sigma(1), \ldots, \sigma(m+m^2)) = (c_1, \ldots, c_m, c_1 c_1, \ldots, c_m c_m)$. The inner product is $\langle f, g \rangle = \int_D f(t) g(t) dt$. Since $\sigma(\mathbf{c})$ is a second order polynomial in the $c_i$, the penalty is actually as fourth order polynomial in the $c_i$.

We can write the Penalised Sum of Squared Errors in terms of the $c_i$

$$PENSSE(\mathbf{c}) = (\mathbf{y} - \mathbf{\Phi c})'(\mathbf{y} - \mathbf{\Phi c}) + \lambda\sigma(\mathbf{c})'\mathbf{K}\sigma(\mathbf{c}).$$

### Alternative Representation Of The Penalty By Splitting the Inner Product

It is difficult to deal with the indexing above. An alternative is to break the function into two parts $f = f' + f^2$ where $f' = \sum c_k\phi_k'$ and $f^2 = \sum c_k c_l \phi_k \phi_l$. Since $\langle f', f^2\rangle = \langle f^2, f'$. We can represent the penalty in the form of the sum of three parts, letting $\mathbf{b} = (c_1c_1, c_1c_2, \dots)$

$$\langle f' + f^2, f' + f^2\rangle = \mathbf{c}'\mathbf{Kc} + 2\mathbf{c}'\mathbf{Lb} + \mathbf{b}'\mathbf{Mb}.$$

Here $\mathbf{K}_{ij} = \langle\phi_i', \phi_j'\rangle$. $\mathbf{L}$ and $\mathbf{M}$ similarly represent $\langle f', f^2\rangle$ and $\langle f^2, f^2\rangle$.

### Using the Vec Operator to Represent the Penalty

**Definition.** The Vec operator applied to a matrix stacks all the matrix's columns on top of each other

We can represent the vector of products $c_ic_j$ as $\operatorname{Vec}\mathbf{cc}'$. Note that each term $c_ic_j$ can appear twice if $i \neq j$. We must therefore scale any inner product matrices appropriately. We can write the penalty without messing with indices

$$\langle f' + f^2, f' + f^2\rangle = \mathbf{c}'\mathbf{Kc} + 2\mathbf{c}'\mathbf{L}\operatorname{Vec}(\mathbf{cc}') + \operatorname{Vec}(\mathbf{cc}')'\mathbf{M}\operatorname{Vec}(\mathbf{cc}')$$

## 7 Systems Of Ordinary Differential Equations (ODES)

The approach generalises quite well to systems of differential equations. In contrast to the multivariate case above, where one output variable depended on multiple input variables, we now have a time series of vectors $\mathbf{x}_k$ depending only on time. We give each component of $\mathbf{x}(t)$ its own basis expansion, but keep the same basis functions. For this section we will concentrate on the two dimensional case, except where it is obvious that we are considering an arbitrary number of dimensions.

$$\begin{aligned}\mathbf{x}_i(t) &= \mathbf{c}_i'\phi(t)\\ &= c_{i1}\phi_i(t) + \cdots + c_{im}\phi_m(t).\end{aligned}$$

### Least Squares Penalty

We will also need an inner product of some sort to define the least squares and roughness penalties. It seems reasonable to use the standard dot product $\mathbf{x} \cdot \mathbf{x}$ in place of $|x|^2$ for our penalties. For a two dimensional series $\mathbf{x} = (x, y)$ with coefficient vectors $\mathbf{b}$ and $\mathbf{c}$, our least squares penalty has the form

$$\begin{aligned} SSE &= \sum_{i=1}^{n}\{[x_i - \sum_{j=1}^{m} b_j\phi_j(t_i)]^2 + [y_i - \sum_{j=1}^{m} c_j\phi_j(t_i)]^2\} \\ &= (\mathbf{x} - \mathbf{\Phi b})'(\mathbf{x} - \mathbf{\Phi b}) + (\mathbf{y} - \mathbf{\Phi c})'(\mathbf{y} - \mathbf{\Phi c}) \\ &= \|[\mathbf{x}\,\mathbf{y}] - \mathbf{\Phi}[\mathbf{b}\,\mathbf{c}]\|_F^2. \end{aligned}$$

$\| \cdot \|_F^2$ is the Frobenius Norm. For an $n \times m$ matrix it is defined in terms of the sums of the squares of its elements

$$\|A\|_F^2 = \sum_{i=1}^{n}\sum_{j=1}^{m} A_{ij}^2.$$

## Differential Equation Penalty

A system of differential equations has the form

$$\begin{aligned} \mathbf{x}' &= f(\mathbf{x}, t) \\ \mathbf{x}(t_0) &= \mathbf{x_0} \end{aligned}$$

We won't be too worried about making sure our penalties are in the standard form above though. It sometimes convenient to leave a higher order derivative on the left hand side instead of converting them to the standard form.

As usual we will be dealing with expressions of the form $\|T\mathbf{x}\|$, except $\mathbf{x}(t)$ is a vector valued function, or a curve. We will use the following $L^2$ inner product to induce our norm

$$\langle \mathbf{x}, \mathbf{y} \rangle = \int_D \mathbf{x}(t) \cdot \mathbf{y}(t)\mathrm{d}t.$$

Thanks to the relative abstractness of an inner product space, we will not have to make too many changes in moving into multivariate data.

**Example 6.** Roughness Penalties

Suppose we have a penalty of the form $\|\mathbf{x}''\|$. If we expand out the inner product we see $\|\mathbf{x}''\|^2 = \sum \int_D \mathbf{x}_k(t)^2\mathrm{d}t$. Since we assume each component of $\mathbf{x}$ has its own basis function expansion, we can use the previous results on roughness penalties to find

$$\|\mathbf{x}\|^2 = \sum \mathbf{c}_i'\mathbf{K}\mathbf{c}_i.$$

Here as usual, $\mathbf{K}_{ij} = \langle \phi_i, \phi_j \rangle$ where we use the one-dimensional inner product $\langle f, g \rangle = \int_D fg\mathrm{d}t$.

Notice that our penalty decomposes into a sum of simpler penalties. This suggests we could take a multiple penalty approach. Instead of an expression of the form $\lambda\|\mathbf{x}\|^2$ we have a sum of penalties

$$\sum \lambda_k\|\mathbf{x}_k\|^2 = \lambda_1\mathbf{c}_1'\mathbf{K}\mathbf{c}_1 + \cdots + \lambda_m\mathbf{c}_m'\mathbf{K}\mathbf{c}_m$$

**Example 7.** Generalised Roughness Penalties

In all the previous cases the weights $\lambda$ have been *external* to the norms we used. What if we instead used a weighted inner product of the form $\langle x, y\rangle_Q = \mathbf{x}'\mathbf{Q}\mathbf{y}$? To define an inner product we must have that $\mathbf{Q}$ be symmetric and positive definite. However since we only have $\lambda \geq 0$ in general, we will only say that $\mathbf{Q}$ must be positive semidefinite and symmetric. We define a symmetric, positive semidefinite bilinear form, but still retain the inner product notation, $\langle \mathbf{x}, \mathbf{y}\rangle = \int_D \mathbf{x}'\mathbf{Q}\mathbf{y}\,dt$.

What will the roughness penalty look like with this change? If $\mathbf{Q}$ is diagonal then we will get the results above with the multiple $\lambda$'s.

In the case of a general suitable matrix, by making use of the usual approach we find

$$\begin{aligned}\|\mathbf{x}''\|_Q^2 &= \sum\sum q_{ij}\langle \mathbf{x}_i'', \mathbf{x}_j''\rangle_{L^2} \\ &= \sum\sum q_{ij}\mathbf{c}_i'\mathbf{K}\mathbf{c}_j.\end{aligned}$$

**Example 8.** Lotka Volterra Equations

The Lotka Volterra Equations are a model of the interactions of a prey and predator population. They are as follows

$$\begin{aligned}x' &= x(\alpha - \beta y) \\ &= \alpha x - \beta xy \\ y' &= -y(\gamma - \delta x) \\ &= -\gamma y - \delta xy.\end{aligned}$$

We will be using the standard dot product for our norms. In the standard inner product, the different terms are independent of each other; we only need to look at one equation , so without loss of generality we will find a formula for

$$\|x' - \alpha x - \beta xy\|^2.$$

Here we are computing a "one dimensional" penalty. Notice we no longer assume anything about the signs of the coefficients.

By the usual methods we see

$$\begin{aligned}\|x' - \alpha x - \beta xy\|^2 &= \langle x' - \alpha x - \beta xy, x' - \alpha x - \beta xy\rangle \\ &= \|x'\|^2 + \alpha^2\|x\|^2 + \beta^2\|xy\|^2 + 2\alpha\beta\langle xy, x\rangle - 2\alpha\langle x', x\rangle - 2\beta\langle x', xy\rangle.\end{aligned}$$

Many of the terms were covered in previous examples, so only examine the non-linear interaction terms bear any serious examination. We will not bother defining the matrices that appear, as their definitions should be clear by now.

Firstly $xy = \sum\sum b_i\phi_i c_j\phi_j$. If we define $d = (b_1c_1, \ldots, b_nc_n)$ we get

$$\|xy\|^2 = \mathbf{d}'\mathbf{K}\mathbf{d}.$$

Where $\mathbf{K}_{ikjl} = \langle \phi_i\phi_k, \phi_j\phi_l \rangle$. We can devise similar results for all the other terms.

We will use the Vec operator again to save space. Let $\mathbf{d} = \text{Vec}(\mathbf{c}\mathbf{b}')$ as before. We can express the penalty analytically

$$\|x' - \alpha x - \beta xy\|^2 = \mathbf{b}'\mathbf{K}\mathbf{b} + \alpha^2\mathbf{b}'\mathbf{L}\mathbf{b} + \beta^2\mathbf{d}'\mathbf{M}'\mathbf{d} + 2\alpha\beta\mathbf{d}'\mathbf{N}\mathbf{b} - 2\alpha\mathbf{b}'\mathbf{O}\mathbf{b} - 2\beta\mathbf{b}'\mathbf{P}\mathbf{d}.$$