

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

February 26, 2013

1 Introduction To Basis Function Expansion Methods

One of the major fields in Statistics is that of fitting functions to data. A very large class of methods are known as *Basis Function Expansion Methods*, where we estimate the output as a weighted sum of basis functions. These models generally take a form as follows:

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

Here we have our finite set of basis functions $\{\phi_k\}$ along with our parameters $\{c_k\}$, $\mathbb{E}[y|x]$ is the conditional expectation of our independent variable y , given our dependent data x . Roughly $\mathbb{E}[y|x]$ can be thought of as our best estimate of the value of y given x .

Basis Function Methods might seem a little abstract, but they are ubiquitous. For example, Simple Linear Regression is an example of a basis function method. 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:

$$\mathbb{E}[y|t] = \hat{y}(t) = c_0 + c_1t$$

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 or the median 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:

$$\hat{y}(t) = c_0 + c_1t + c_2t^2$$

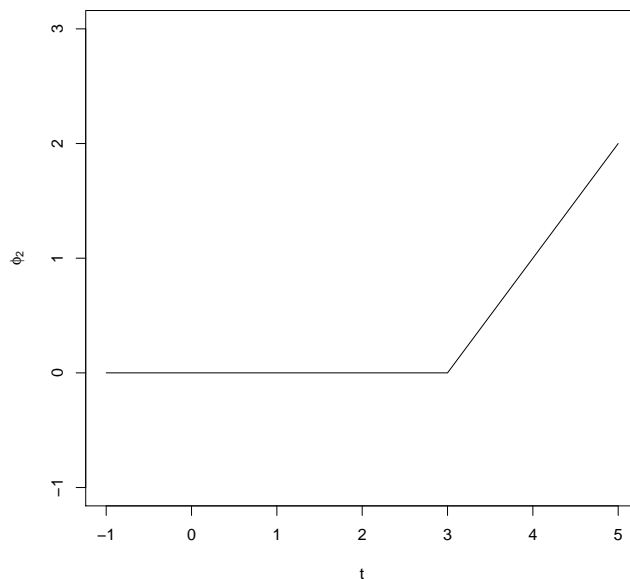


Figure 1: The Ramp Function ϕ_2

There are other choices of basis besides monomials we could use. We could have a basis of two functions $\{\phi_1, \phi_2\}$ Where the two functions are defined as follows:

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

The first function is a constant function, the second is a ramp function based at τ .

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 $\frac{dy}{dt}$. Secondly it is very arbitrary, it is not obvious why it would be useful

This underscores an important point. Real Analysis tells us that there are many sets of functions that are *dense* on some interval $[a, b]$, that is they can approximate any function on $[a, b]$ to arbitrary accuracy. We must be sensible in our 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 simple linear regression and quadratic

regression are useful; they can capture much variation in the observed data in spite of being very simple.

1.1 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 would be wise 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 ω is the frequency. The frequency depends on our time scale, they are related by the formula $\omega = \frac{2\pi}{T}$, 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.

1.2 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 $\{t_0, \dots, t_K\}$.

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 they assume less about the form of the data, they can help us avoid bias.

2 Least Squares Fitting

Least Squares is one of the most well known statistical estimation techniques. If we have a set of n observations y_i , measured at times t_i , then the *Least Squares Criterion*, chooses the estimated function \hat{y} from some set of functions S that minimises the sum of the squares of the error:

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

We assume S is the span of 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} we only need to estimate the coefficients \hat{c}_i . Then we have completely determined \hat{y} . We can then write the least squares criterion as:

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

The above expression is cumbersome, we can use vector notation to simplify it. Firstly, we have $f(t) = \mathbf{c}'\phi(\mathbf{t})$, where $\mathbf{c} = (c_1, \dots, c_m)$ and $\phi = (\phi_1(t), \dots, \phi_m(t))'$. If we construct a matrix Φ , where the i th row of Φ is $\phi(t_i)$, and let $\mathbf{y} = (y_1, \dots, y_n)$ we can write the least squares problem as:

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

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.

3 Roughness Penalties

It is well known that the Least Squares gives us the *Best Linear Unbiased Estimator* for \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 *variance* of \hat{y} and helps guard against overfitting. In the context of basis function expansions, the most commonly used penalty is the curvature penalty:

$$PEN(\hat{y}) = \int_D [\hat{y}(t)']^2 dt$$

Here D is our domain of interest and f'' stands for the second derivative of the function f . This criterion for fitting a curve is very appealing in many regards. It can be shown using the theory of Finite Elements that if we constrain \hat{y} to pass through two points, then the \hat{y} minimises the penalty is the best approximation to the straight line going through the points that we can form with this basis.

This penalty can also be represented as a polynomial. Let $\langle f, g \rangle = \int_D f(t)g(t)dt$. We can see that $\langle \cdot, \cdot \rangle$ defines an inner product on $L^2(D)$, the set of square integrable functions on D . The penalty can be written in the form $PEN(f) = \langle f'', f'' \rangle$. Substituting in the expansion for f we get:

$$\langle f'', f'' \rangle = \left\langle \sum_{i=1}^m c_i \phi_i'', \sum_{i=1}^m c_i \phi_i'' \right\rangle = \sum_{i=1}^m \sum_{j=1}^m \langle c_i \phi_i'', c_j \phi_j'' \rangle = \sum_{i=1}^m \sum_{j=1}^m c_i c_j \langle \phi_i'', \phi_j'' \rangle$$

We can see that the terms $\langle \phi_i'', \phi_j'' \rangle$ depend 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 ϕ_i into a linear combination ϕ_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 ϕ_i'' and have the \mathbf{c} in place of the \mathbf{b} because differentiation is linear.

4 Penalised Least Squares

A roughness penalty on its own is not very useful; it does not make any use of the data at all. Therefore it is more common to use a hybrid approach. Instead of a least squares penalty, we have a *Penalised Sum Of Squared Errors*.

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

If we assume again that f is the sum of basis functions, depending on a vector of coefficients $\hat{\mathbf{c}}$; we can use the previous results to show that the Penalised Sum of Squares can be written as the sum of two quadratic forms:

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

Where \mathbf{y} , \mathbf{K} and Φ have the same definitions as they did before.

5 Multivariate Splines

What if our data is spatial in nature? In this case our data will often be at a series of points $\mathbf{x}_i = (x_i, y_i); i = 1, \dots, n$. 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 estimated function \hat{y} as a basis function expansion:

$$\hat{y}(\mathbf{y}) = \sum_{i=1}^m c_i \phi_i(\mathbf{y})$$

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.

5.1 Fitting the Displacement of a Membrane Using a Laplacian Penalty

One model of the mechanics of a membrane is the emphwave 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 Δ 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*:

$$\Delta u = 0$$

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

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

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

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

Then the laplacian penalty can be written as

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

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 Φ as before with $\Phi_{ij} = \phi_j(\mathbf{x}_i)$, $\mathbf{c} = (c_1, \dots, c_m)$. We can then write the penalty in terms of the coefficients:

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

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

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

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,

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

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 $a_1 \phi_1'' + \dots + a_m \phi_m'' + b_{11} \phi_1 \phi_1 + \dots + b_{mm} \phi_m \phi_m$. 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 ϕ_i'' and the $\phi_i \phi_j$ together. We would define a function $\pi(n)$ that returns the appropriate function either some ϕ_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 $\sigma(\mathbf{c}) = (\sigma(1), \dots, \sigma(m+m^2)) = (c_1, \dots, c_m, c_1c_1, \dots, c_m c_m)$. 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 :

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