Chapter 2

Overview of Existing Methods.

This chapter contains a brief summary of the key methods available for univariate nonparametric regression function estimation from independent data. The goal of these methods is estimation of the regression function without assuming any underlying parametric model. One method in particular, penalized regression with model-based penalties, will be discussed in greater detail. However it is helpful to have a survey of methods before spending more time on any specific method.

First of all, in order to put smoothing problems into a statistical framework, let us describe the two most commonly used models. The first one is the "fixed design", where the ordinates of the data points are thought of as being deterministic values. These ordinates values are usually chosen by the experimenter, as in a designed experiment. Frequently, these points are taken to be equally spaced since it is what one might intuitively think of. The fixed design model will be used in this chapter and throughout this thesis. For reference, the second model is the “random design”, where the data points are thought of as being realizations from a bivariate probability distribution. These ordinates are usually not chosen by the experimenter, and occur, for example, in sample surveys.

Let the x_i's have a fixed design satisfying the following regularity conditions. There exists a continuous positive density f on $[0, 1]$ with

$$\int_0^{x_i} f(t)\, dt = \frac{i}{n+1} \tag{2.1}$$

One can show that there exists $C_1 > 0$ and C_2 such that

$$\frac{C_1}{n+1} \leq x_i - x_{i-1} \leq \frac{C_2}{n+1} \tag{2.2}$$

for all n and all x_i, $i = 1, \ldots, n+1$ (take $x_0 = 0$ and $x_{n+1} = 1$).

The fixed design model is given by

$$Y_i = m(x_i) + \varepsilon_i \tag{2.3}$$

for $i = 1, 2, \ldots, n$, where m is the “mean” or regression function, the x_i's are nonrandom design points satisfying (2.1), and the ε_i's are independent random variables with mean 0 and variance σ^2.

The goal is to use $Y_1, \ldots, Y_n$ to estimate the regression function m or its r^{th} derivative $m^{(r)}$.

The performance of an estimator $\hat{m}_r(x)$ of $m^{(r)}(x)$ is conveniently assessed via its *Mean Squared Error* (MSE) or its *Mean Integrated Squared Error* (MISE) defined respectively by

$$MSE(x) = E\left[\{\hat{m}_r(x) - m^{(r)}(x)\}^2\right], \tag{2.4}$$

$$MISE = \int MSE(x)\omega(x)\, dx \tag{2.5}$$

with $\omega \geq 0$, a weight function. When the MSE-criterion is used, it is understood that the main objective is to estimate the function $m^{(r)}(\cdot)$ at the point x, and when the MISE-criterion is used the main goal is to recover the whole curve. It can be easily seen that the MSE has the following bias-variance decomposition:

$$MSE(x) = \left[E\{\hat{m}_r(x)\} - m^{(r)}(x)\right]^2 + Var\{\hat{m}_r(x)\}. \tag{2.6}$$

One often refers to the term $E\{\hat{m}_r(x)\} - m^{(r)}(x)$ as the *bias*. We will come back to the notion of bias-variance later in section 2.1.3.

2.1 Kernel Estimators

2.1.1 Local Constant Fitting

Nadaraya-Watson Estimator

Without any assumption on the form of the regression function m, a data point far away from x carries very little information about the actual value of $m(x)$. Thus, a reasonable estimator for the mean function m is a running local average and is generally called the locally weighted average estimator. This estimator is a weighted average of the Y_i's with weights depending on a function K and a positive number h. Let K be a given real-valued function to calculate the weights assigned to the observations. This function K is usually a probability density function symmetric about 0 and is often called a kernel function.

The following notation about kernels will be used from now on. The moments of K are denoted by $\nu_j(K) = \int s^j K(s)\, ds$. Gasser, Müller and Mammitzsch (1985) defined K as a kernel of order p_0, p_1 ($p_0 \geq 0, p_1 \geq p_0 + 2$) if $\nu_j(K) = 0, j = 0, \ldots, p_0 - 1, j = p_0 + 1, \ldots, p_1 - 1, \nu_{p_0}(K) \neq 0$, and $\nu_{p_1}(K)$ is finite and nonzero. Generally, p_0 is taken to be the order of the derivative of the regression function m to be estimated while $p_1 - p_0$ controls the amplitude of the estimation bias. We say that K is a kernel of order p if it is a kernel of order $p_0 = 0, p_1 = p$.

Let h be a bandwidth (also called a smoothing parameter), which is a non-negative number controlling the width of the neighborhood in which averaging is performed. Then, the well-known *Nadaraya-Watson* (N-W from now on) kernel regression estimator is given by:

$$\hat{m}_{NW}(x) = \frac{\sum_{i=1}^{n} Y_i K(\frac{x_i - x}{h})}{\sum_{i=1}^{n} K(\frac{x_i - x}{h})} \tag{2.7}$$

See Nadaraya (1964) and Watson (1964) for more details.

Some commonly used kernel functions include the Gaussian kernel $K(t) = (2\pi)^{-\frac{1}{2}} \exp(-t^2/2)$ and the symmetric *Beta* family kernel

$$K(t) = \frac{(1-t^2)_+^{\gamma}}{Beta(\frac{1}{2}, \gamma + 1)} \qquad \gamma = 0, 1, \ldots$$

where the subscript $+$ denotes the positive part. The term $Beta(\alpha, \beta)$ denotes the beta function,

$$Beta(\alpha, \beta) = \int_0^1 x^{\alpha - 1}(1-x)^{\beta - 1}\, dx.$$

The beta function is related to the gamma function through the following identity:

$$Beta(\alpha, \beta) = \frac{\Gamma(\alpha)\Gamma(\beta)}{\Gamma(\alpha+\beta)}.$$

The choice of $\gamma = 0, 1, 2, 3$ leads respectively to the uniform, the Epanechnikov, the biweight and the triweight kernel function. Marron and Nolan (1988) stated that the Beta family includes the most widely used kernel functions, the Gaussian kernel being the limit as $\gamma \to \infty$.

It is easily shown that the N-W estimate is the solution to a natural weighted least squares problem, being

$$\begin{aligned} \hat{m}_h(x) &= argmin_\beta \sum_{i=1}^{n} (Y_i - \beta)^2 w_i \\ &= \frac{\sum_{i=1}^{n} w_i Y_i}{\sum_{i=1}^{n} w_i} \end{aligned} \tag{2.8}$$

where the weights w_i are $K((x_i - x)/h)$. That is, the N-W estimator corresponds to locally approximating $m(x)$ with a constant, weighting values of Y corresponding to x_i's closer to x more heavily.

Note that one can estimate the r^{th} derivative of m by differentiating $\hat{m}(x)$ in (2.8), or one can use the asymptotically equivalent estimate (for equally spaced design)

$$\widehat{m^{(r)}}(x) = \frac{1}{(n+1)h^{r+1}} \sum_{i=1}^{n} Y_i K^\star\left(\frac{x_i - x}{h}\right), \tag{2.9}$$

with $K^\star(u) = (-1)^{(r)} K^{(r)}(u)/C$ where C is so that $\nu_r(K^\star) = r!$. One can show that if K is of order p, then $K^\star$ is a kernel of order $r, r+p$.

Gasser-Müller Estimator

Another estimator using local constant approximation is the *Gasser-Müller* (G-M from now on) estimator and is defined as

$$\hat{m}_{GM}(x) = h^{-1} \sum_{i=1}^{n} \left[\int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right) du \right] Y_i, \tag{2.10}$$

where $x_{i-1} \leq s_{i-1} \leq x_i$ (a common choice being $s_{i-1} = (x_{i-1} + x_i)/2$, with s_0 and s_n being the upper and lower limits of the range of x, respectively). When restricted to $x \in [h, 1-h]$, $K(u) = 0$ for $u \notin [-1, 1]$ and K integrable to 1, the weights $w_i = h^{-1} \int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right) du$ sum to 1, and hence, the G-M estimator is of the form (2.8). Often, estimators of this type are called *convolution estimators*.

The G-M estimate can be viewed as a "continuous moving local average" as the weights $h^{-1} \int_{s_{i-1}}^{s_i} K\left(\frac{x-u}{h}\right) du$ are continuous functions of x, even if K isn't. On the other hand, the weights used in the N-W estimator could be discontinuous. If the design points are equally spaced, then the N-W and the G-M estimators are very nearly the same, by the integral mean value theorem. However, when the design points are not equally spaced, or when the design points are iid random variables, there are substantial and important differences between these two estimators. These cases are not of main interest here and therefore will not be covered in detail in this survey.

2.1.2 Local Polynomial Fitting

The local constant Nadaraya-Watson estimator can be modified to an estimator that fits higher order local polynomials. This might be desirable since a high order polynomial may better approximate a function locally.

A lot of work has been done on local polynomial regression. It was studied by Stone (1977), Cleveland (1979) and Stone (1980,1982). Some more recent work includes Fan (1992,1993), Fan and Gijbels (1992) and Ruppert and Wand (1994). A detailed picture of the attractive properties of local polynomial fitting is given in these papers.

Suppose that locally the regression function m can be approximated by

$$m(z) \approx \sum_{j=0}^{p} \frac{m^{(j)}(x)}{j!}(z-x)^j \equiv \sum_{j=0}^{p} \beta_j (z-x)^j \tag{2.11}$$

for z in a neighborhood of x, using Taylor's expansion. Because we estimate $m(z)$ locally by a polynomial model, we are inclined to use a locally weighted polynomial regression

$$\sum_{i=1}^{n} \{Y_i - \sum_{j=0}^{p} \beta_j (x_j - x)^j\}^2 K\left(\frac{x_i - x}{h}\right) \tag{2.12}$$

where $K(\cdot)$ is a kernel function as defined previously and h is the smoothing parameter. Let $\hat{\beta}_j$, $j = 0, \ldots, p$ be the minimizer of (2.12). Then an estimator for $m^{(r)}(x)$ is

$$\hat{m}_r(x) = r!\hat{\beta}_r. \tag{2.13}$$

One might think that this is pretty similar to the traditional parametric approach where the regression function is *globally* modeled by a polynomial. However, to have an acceptable modeling bias, the degree p of the globally fitted polynomial is often required to be quite large. For example, for a simple function such as $m(x) = \sin x + \cos x, x \in [0, 2\pi]$, parametric polynomial modeling requires a large p in order to get a low bias. This large degree p

introduces an over-parameterization, causing the estimated parameters to be highly variable. A consequence of this is a large variance of the estimated regression function. Local polynomial regression being a "local" method, it only requires a small degree p of the polynomial fitted locally. Usually a degree of $p = r+1$ or $p = r+3$ is used (where r represents the order of the derivative to be estimated). For example, using a polynomial of degree $p = 1$ leads to a very popular estimator $\hat{m}(x)$ called the local linear fit or the local linear regression smoother.

In the case of fixed design, the N-W estimate, the G-M estimate and the local linear estimate all have the same asymptotic bias and variance expressions when $x \neq 0$ or 1. These are respectively

$$\begin{aligned} Asympt.Bias &= \frac{1}{2}h^2 m''(x)\nu_2(K) + O(\frac{1}{\sqrt{nh}}) + o(h^2) && (2.14) \\ Asympt.Variance &= \frac{\sigma^2}{nhf(x)}\nu_0(K^2) + O(\frac{1}{n^2h^2}). && (2.15) \end{aligned}$$

However, in the case of a random design they perform differently. See Table 2.1 for the asymptotic expressions in the random design case. Comparison of these estimates is discussed in greater detail in Eubank (1988), Müller (1988), Härdle (1990), Fan (1992) and Cheng, Fan and Marron (1997).

2.1.3 Bias and Variance Tradeoff for Polynomial Fitting

A natural question is how do we proceed in the selection of the smoothing parameter. Choosing a "good" bandwidth is crucial to obtain a sensible curve

Table 2.1: *Asymptotic bias and variance of kernel regression smoothers in the case of random design. Taken from Fan (1992).*

Method	Bias	Variance
N-W	$(m''(x) + \frac{2m'(x)f'(x)}{f(x)})b_n$	V_n
G-M	$m''(x)b_n$	$1.5V_n$
Local Linear	$m''(x)b_n$	V_n

where $b_n = \frac{1}{2}h^2\nu_2(K)$ and $V_n = \frac{\sigma^2}{f(x)nh}\nu_0(K^2)$.

estimate. As mentioned earlier, the smoothing parameter is the value that determines the width of the neighborhood where the local estimate will be computed. One can choose h "by eye" (pretty feasible when $x \in \mathcal{R}$) which means that one chooses the width of the neighborhood by looking at plots of the estimate for different values of h. By doing this, one can assure oneself that the resulting estimate of m is smooth enough while tracking the data pretty well.

Also, the bandwidth can be chosen automatically using some well known techniques like cross-validation and plug-in methods that will be introduced shortly. However, automatic techniques appear more desirable since they are data driven, and thus somewhat objective. We will discuss automatic bandwidth selectors in Chapter 3. Nevertheless, even with automatic bandwidth choice, further tuning by the data analyst is often beneficial.

Let us demonstrate the influence of bandwidth selection on the estimated regression function by two examples taken in the context of local linear regression (see Table 2.2).

Table 2.2: *Effect of bandwidth selection on the local linear regression estimator.*

Small Bandwidth: $h \to 0$	Large Bandwidth: $h \to \infty$
Window contains only 1 x_i	Window contains all x_i's
$\hat{m}_h(x_i) = y_i$	$\hat{m}_h(x_i) = \hat{\alpha} + \hat{\beta}x_i$
$E(\hat{m}_h(x_i)) = E(y_i) = m(x_i)$	$E(\hat{m}_h(x_i)) \neq m(x_i)$
$Var(\hat{m}_h(x_i)) = Var(y_i) = \sigma^2$	$Var(\hat{m}_h(x_i)) = \sigma^2(\frac{1}{n} + \frac{(x_i-\bar{x})^2}{\sum_{i=1}^{n}(x_i-\bar{x})^2})$
$\not\to 0$ as $n \to \infty$	$\to 0$ as $n \to \infty$

Consider a small bandwidth ($h \to 0$) and a very large one ($h \to \infty$). Obviously taking h very close to 0 gives a window that only contains one x_i, which results in an estimate interpolating the data. Using $h \to \infty$ gives a window that contains all the x_i's which is then equivalent to fitting a global linear model $m(x) = \alpha + \beta x$. An estimate using a small bandwidth does not suffer from bias but does suffer from high variability. On the contrary, an estimate with a large bandwidth is usually biased but has a low variance. The problem is then to find a compromise, i.e., a smoothing parameter so that both the bias and the variance are minimal. This problem is often referred to as the "Bias-Variance Tradeoff". Since this forms the core of many bandwidth selection criteria, it is essential to have a good understanding of the bias and variance of the estimators. We will, in this section, present the asymptotic expressions for the bias and variance of the local polynomial estimate (degree p) of the rth derivative of the regression function m.

Let N_p be the $(p+1)$x$(p+1)$ matrix having (i, j) entry equal to $\nu_{i+j-2}(K)$ and let $M_{r,p}(u)$ be the same as N_p, but with the $(r+1)$st column replaced by

$(1, u, \ldots, u^p)^T$.

Ruppert and Wand (1994) defined the kernel

$$K_{(r,p)}(u) = r!\{|M_{r,p}|/|N_p|\}K(u), \tag{2.16}$$

where $K(u)$ is a symmetric density. It can be shown that $(-1)^r K_{(r,p)}$ is an order (r, s) kernel as defined by Gasser, Müller and Mammitzsch (1985) where $s = p + 1$ when $p - r$ is odd and $s = p + 2$ when $p - r$ is even. Kernels of that form are known to perform well when estimating the r^{th} derivative of a regression function.

The asymptotic expansions for the bias and variance of the estimator $\hat{m}_r(x)$ are given in the next theorem. These results are taken from Ruppert and Wand (1994) and were derived for the random design case. However, similar results also hold for the fixed design case. We will first introduce the original results for a general local p^{th} degree polynomial fit and derivative r.

First, let's assume the following.

1. The X_i's are iid random observations following the density f.

2. The kernel K is a compactly supported, bounded kernel with $\nu_2(K) \neq 0$. In addition, all odd-order moments of K are equal to zero.

3. The point x is in the support of f. At x, f is continuously differentiable and $m^{(p+2)}$ is continuous in a neighborhood of x. Also, $f(x) > 0$ and $\sigma^2 > 0$.

Theorem 1 *Suppose that x is an interior point of the support of f and that*

(1), (2) and (3) hold. Further, assume that $h \to 0$, $nh^{2r+1} \to \infty$ as $n \to \infty$. Then, for $p-r$ odd,

$$Bias\{\hat{m}_r(x)|X_1,\ldots,X_n\} = \nu_{p+1}(K_{(r,p)})\left(\frac{m^{(p+1)}(x)}{(p+1)!}\right)h^{p-r+1} + o_p(h^{p-r+1}) \tag{2.17}$$

and, for $p-r$ even,

$$\begin{aligned} Bias\{\hat{m}_r(x)|X_1,\ldots,X_n\} &= h^{p-r+2}\Bigg[\nu_{p+2}(K_{(r,p)})\left(\frac{m^{(p+2)}(x)}{(p+2)!}\right) \\ &+ \left\{\nu_{p+2}(K_{(r,p)}) - r\nu_{p+1}(K_{(r-1,p)})\right\}\frac{m^{(p+1)}(x)f'(x)}{(p+1)!f(x)}\Bigg] \\ &+ o_p(h^{p-r+2}). \end{aligned} \tag{2.18}$$

In both cases

$$Var\{\hat{m}_r(x)|X_1,\ldots,X_n\} = \nu_0(K^2_{(r,p)})\frac{\sigma^2}{nh^{2r+1}f(x)}\{1+o_p(1)\}. \tag{2.19}$$

It is clear from this Theorem that the case of $p-r$ odd exhibits a much simpler bias expression. In fact, unlike the case of $p-r$ even which depends on $m^{(p+1)}(x)$, $m^{(p+2)}(x)$ and $f'(x)/f(x)$, it only depends on $m^{(p+1)}(x)$.

It can be shown that the bias of $\hat{m}_r(x)$ at the boundary is of the same order as the interior bias when $p-r$ is odd and is of higher degree when $p-r$ is even. See Ruppert and Wand (1994) for more details on the subject.

The results for the fixed design case are the same, and can be derived by techniques similar to those used by Li (1996). However, we need to assume the following along with assumptions 2 and 3 of Theorem 1:

1. The x_i's follow a fixed design with density f.

2. The first derivative of the kernel density K is continuous.

2.1.4 Bandwidth Selection for Local Polynomial Regression

Now that the concept of bias-variance tradeoff has been explained, it is time to introduce bandwidth selection techniques. The main topic of this work is smoothing parameter selection with smoothing splines for correlated errors. This chapter will only survey a few commonly used methods in local polynomial regression and smoothing splines when the errors are independent and Chapter 3 will describe in greater detail the methods for correlated errors.

As mentioned earlier, the appearance of a local polynomial estimate depends strongly on the value of the bandwidth h. The two most popular automatic techniques are known as *cross-validation* (CV) and *plug-in* methods.

Cross-validation uses the principle of "leave-one-out" prediction. The idea is to leave the data points out one at a time and to select the value of the bandwidth under which the removed data points are best predicted by the remaining data. The cross-validation criterion to be minimized is the following

$$CV(h) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{m}^{(i)}(x_i)\right)^2 \tag{2.20}$$

where $\hat{m}^{(i)}(x_i)$ is the estimate, using bandwidth h, based on the data with x_i, Y_i removed. The minimizer of CV, $\hat{h}_{CV}$, is the cross-validation choice of the optimal bandwidth value. Unfortunately, studies have shown (see Hall and Marron (1987), Park and Marron (1990)) that this technique's theoretical and practical performances are disappointing. In particular, it leads to an estimate of h that suffers from high variance.

Let us now describe the *Direct Plug-in* method of Ruppert, Sheather and Wand (1995). Consider the MISE criterion as seen in (2.5) where $w = f$, the design density, and $r = 0$. Then, the bandwidth that minimizes (2.5), and thus asymptotically of (2.20), is the following

$$h_{opt} = \left[\frac{(p+1)(p!)^2\nu_0(K^2)\sigma^2}{2n\nu_{p+1}^2(K)\int m^{(p+1)}(u)^2 f(u)\,du}\right]^{1/(2p+3)} \tag{2.21}$$

using a local polynomial of odd degree p and a kernel K satisfying assumption 2 of Theorem 1, provided $\int m^{(p+1)}(u)^2 f(u)\,du$ is nonzero. The plug-in bandwidth selector requires an estimate of σ^2 and $\int m^{(p+1)}(u)^2 f(u)\,du$.

For sake of simplicity, let us assume that we are estimating m via a local linear ($p = 1$) kernel regression method. Then, equation (2.21) simplifies to

$$h_{opt} = \left[\frac{\nu_0(K^2)\sigma^2}{n\nu_2^2(K)\int m''(u)^2 f(u)\,du}\right]^{1/5}. \tag{2.22}$$

However, σ^2 and $\int m''(u)^2 f(u)\,du$ are unknown so we need to find an estimate of those two quantities in order to find h_{opt}.

There exists a relatively extensive literature on the estimation of σ^2 in the homoscedastic nonparametric regression context. Ruppert, Sheather and Wand (1995) extended the idea of Hall and Marron (1990) based on a zero-degree least squares fit to a general p^{th} degree setting. This leads to an estimate of the form:

$$\hat{\sigma}^2(\lambda) = \nu^{-1}\sum_{i=1}^{n}(Y_i - \hat{m}(x_i))^2 \tag{2.23}$$

where

$$\nu = n - 2\sum_{i=1}^{n} w_{ii} + \sum_{i=1}^{n}\sum_{j=1}^{n} w_{ij}^2, \tag{2.24}$$

$$w_{ij} = \mathbf{e}_1^T(\mathbf{X}_{p,x_i}^T \mathbf{W}_{x_i} \mathbf{X}_{p,x_i})^{-1} \mathbf{X}_{p,x_i}^T \mathbf{W}_{x_i} \mathbf{e}_j \tag{2.25}$$

and $\mathbf{X}_{p,x}$ is the nx$(p+1)$ design matrix

$$\begin{bmatrix} 1 & x_1 - x & \cdots & (x_1 - x)^p \\ \vdots & \vdots & \ddots & \vdots \\ 1 & x_n - x & \cdots & (x_n - x)^p \end{bmatrix}, \tag{2.26}$$

$\mathbf{W}_x = diag[K\{(x_1 - x)/h\}/h, \ldots, K\{(x_n - x)/h\}/h]$ and $\mathbf{e}_j$ is the $(p+1)$x1 vector having 1 in the j^{th} entry and zero elsewhere. For more details on how to compute an estimate of σ^2, please refer to the above mentioned paper.

In order to estimate h_{opt}, one must also estimate the integral in the denominator of (2.22). We saw earlier in equation (2.13) that it is possible to find an estimate of $\hat{m}''(x)$ when using local cubic polynomial regression. Using such an estimate, the bandwidth h_1 that minimises the $MISE$ criterion for $\hat{m}''(x)$ is given by:

$$h_1^{opt} = Dn^{-1/9} \tag{2.27}$$

where D is a function of σ^2 and $m^{(4)}$. So now, a new bandwidth must be chosen.

One can use the following steps in order to get the "RULE OF THUMB" estimate of h_{opt} as proposed by Ruppert, Sheather and Wand (1995).

1. Find the estimate $\hat{\sigma}^2$ of σ^2, as, for instance, in (2.23).

2. Use a parametric family to estimate m by $\hat{m}_P$. Take the 4^{th} derivative to obtain $\hat{m}_P^{(4)}$. Estimate $\int [m^{(4)}(x)]^2 \, dx$ by $\int [\hat{m}_P^{(4)}(x)]^2 \, dx$.

3. Use the estimates in 1 and 2 in the formula (2.27) to estimate h_1^{opt}.

4. Estimate m'' using a local cubic polynomial and the estimate of h_1^{opt}.

5. Use 4 to estimate $\int m''(x)^2 f(x) \, dx$ and plug this estimate along with the estimate of σ^2 into (2.22) to get an estimate of h_{opt}.

For more details on plug-in bandwidth selection methods, please refer to Ruppert, Sheather and Wand (1995).

2.2 Splines

We saw in the previous section that the use of local methods such as local polynomial fitting provided some added flexibility in estimating the regression function. In this section, a different way of enhancing the flexibility of the regression method will be introduced. This method is known as the 'spline method'. Eubank (1988), Wahba (1990) and Green and Silverman (1994) constitute excellent reference books on spline smoothing.

The function ϕ is called a B-spline of degree D defined on the interval $[a, b]$ with knots $t_1, \ldots, t_L$ $(a < t_1 < \cdots < t_L < b)$ if ϕ is a polynomial of degree D on the subintervals $[a, t_1], [t_1, t_2], \cdots, [t_L, b]$ and ϕ has $D - 1$ continuous derivatives on $[a, b]$. We denote the family of these B-splines by $BSp(t_1, \cdots, t_L; D)$.The number of parameters required to specify a B-spline is

$L+D+1$. It is possible to show that $BSp(t_1, \cdots, t_L; D)$ is a linear space. This means that if ϕ_1 and $\phi_2 \in BSp(t_1, \cdots, t_L; D)$ and α_1 and α_2 are real numbers, then $\alpha_1\phi_1 + \alpha_2\phi_2 \in BSp(t_1, \cdots, t_L; D)$.

We also say that ϕ is a univariate natural polynomial spline of degree $D = 2M - 1$ with knots $t_1, \ldots, t_L$ $(a < t_1 < \cdots < t_L < b)$ if $\phi \in BSp(t_1, \cdots, t_L; D)$ and $\phi^{(M)}(t) = 0$ for $t \in [a, t_1] \cup [t_L, b]$.

Let us demonstrate the previous definitions by a simple example where we use cubic splines ($D = 3$). A cubic B-spline is cubic on each of the subintervals and has a continuous second derivative. A natural cubic spline is also a cubic B-spline except that it is linear on the interval $[a, t_1] \cup [t_L, b]$. Cubic splines are the most commonly used splines since polynomials of degree higher than three are too wiggly while lower degree polynomials are not flexible enough.

Now that we have introduced the concept of splines, we will discuss the two most important methods of fitting splines to a model as in (2.3) where m is known to be a smooth function. These are respectively regression splines and smoothing splines.

2.2.1 Regression Splines

In order to discuss regression splines, we need to define a basis for $BSp(t_1, \cdots, t_L; D)$. Let

$$\ldots = t_{-1} = t_0 = a < t_1 < \cdots < t_L < b = t_{L+1} = t_{L+2} = \ldots \tag{2.28}$$

We define the B-spline basis for $BSp(t_1, \cdots, t_L; D)$ by $\{N_{-D,D}, \cdots, N_{L,D}\}$ where the $N_{j,D}$'s are defined recursively by

$$N_{i,D}(x) = \frac{x - t_i}{t_{i+D} - t_i} N_{i,D-1}(x) + \frac{t_{i+D+1} - x}{t_{i+D+1} - t_{i+1}} N_{i+1,D-1}(x), \tag{2.29}$$

given $N_{i,D-1}$ and $i = -(D-1), \ldots, L$. We define $0/0 = 0$, undefined$/0 = 0$ and

$$\begin{aligned} N_{i,0}(x) &= I\{x \in [t_i, t_{i+1})\} \quad i = 0, \cdots, L-1 \\ & N_{L,0}(x) = I\{x \in [t_L, b]\}. \end{aligned} \tag{2.30}$$

So, given D and a set of knots $t_{-(D-1)}, \ldots, t_L$, we have a basis of dimension $D + L + 1$ that defines a linear space that may be used as regression functions when wanting to fit a smooth function m.

Thus, one can estimate $m(x)$ as $\sum_j \theta_j N_{j,D}(x)$. The spline method consists of finding the best spline estimate in a least squares problem. Let $\hat{\theta}$ be the least squares estimate. Then $m(x)$ is estimated by $\hat{m}(x) = \sum_j \hat{\theta}_j N_{j,D}(x)$.

Obviously this method is very dependent on the number and on the location of the knots. An automated method of knot selection is therefore desirable. Introduced by Smith (1982) and Breiman, Friedman, Olshen and Stone (1993) the procedure known as "knot deletion" is a popular automated procedure but will not be discussed here. Instead, a related method which avoids knot selection and is more relevant to the topic of this thesis will be discussed. This procedure relies on the *smoothing spline* approach, see Whittaker (1923), Wahba (1990) or Green and Silverman (1994).

2.2.2 Smoothing Splines

First, let us consider a simple least squares problem:

$$min_m \sum_{i=1}^{n} \{Y_i - m(x_i)\}^2 . \tag{2.31}$$

Obviously the solution of this rather naive least squares problem is any function m which will interpolate the data. However this solution is not desirable in a statistical context. Not only does it produce a highly complex model (with the same dimension as the data set) but it results in an over-parametrisation which leads to a large variability of the estimated parameters. What could be done in order to improve this approach? The main problem is that we did not penalize for the over-parametrisation or interpolation. A popular way to deal with this problem is to penalize via the roughness of the estimate. The roughness is measured by $P(m) = \int (Lm(x))^2 \, dx$, where L is a differential operator which will be discussed in greater detail later. By adding this penalty term to our simple least squares problem we now have the following minimization problem:

$$\hat{m} = min_m \sum_{i=1}^{n} \{Y_i - m(x_i)\}^2 + \lambda P(m) \tag{2.32}$$

where λ is a non-negative real number called the smoothing parameter. So $\hat{m}$ is then, for a given value of λ, the best compromise between smoothness and goodness of fit.

Also, it is possible to show that the minimizer of (2.32) with $P(m) = \int (m^{(M)}(x))^2 \, dx$, is a *natural spline* of degree $2M - 1$ with knots $\{x_i\}$. In this case, one can show (see equation (2.2) of Wahba (1975)) that $\hat{m}$ is linear in the observations Y_i, in the sense that there exists a weight function $G(s,t)$ such that

$$\hat{m}(s) = n^{-1} \sum_{i=1}^{n} Y_i G(s, x_i). \tag{2.33}$$

The weight function depends on the design points $x_1, \ldots, x_n$ and also on the smoothing parameter. The results in (2.33) also hold for the penalty functions described in section 2.2.3, equations (2.35) and (2.36).

As one can notice, (2.32) consists of two distinct parts. The first part imposes a penalty on the lack of fit as we saw in the above least squares problem. The latter part, as mentioned earlier, adds a penalty on the roughness of the estimating function, thus relating to over-parametrisation.

It becomes clear that, in (2.32), the value of the smoothing parameter λ controls the complexity of the model. For example, let us consider a λ of value 0. In that case we fall back to the minimization problem of the form (2.31), which inevitably results in an estimate that interpolates the data. On the other hand, the case where λ is very large leads to an estimate of m converging to the least squares estimate restricted to the kernel of P, that is, the set of m with $P(m) = 0$. In the cubic spline case, when $P(m) = \int (m''(x))^2 \, dx$, the kernel of P is the family of lines.

In fact, if the data Y_i lie on a curve in the kernel of P, then the estimated regression function $\hat{m}$ will be exactly that curve. Thus, we can see that through its kernel, the penalty P is related to a favoured parametric model. Thus, for example, the parametric model $m(x) = \beta_0 + \beta_1 x$ may be considered the favoured model for the penalty $\int (m''(x))^2 \, dx$.

It is important to note that this relationship is not very strong: nothing forbids the estimate $\hat{m}$ to be quite far from the favoured parametric model. So the choice of the roughness penalty may appear to have little importance on the estimation. However it was shown in Heckman and Ramsay (1996) that the choice of an appropriate penalty can lead to better estimates. In fact, as seen in Wahba (1990), the minimizer $\hat{m}$ of (2.32) follows this inequality:

$$\frac{1}{n}\sum_{i=1}^{n}\left(E(\hat{m}(x_i)) - m(x_i)\right)^2 \leq \lambda P(m) \tag{2.34}$$

where m is the true regression function. It is then clear that the closer m is to the kernel of the penalty, the smaller the bias of the resulting estimate will be. Therefore, if the data analyst has some knowledge about the family of curves to which the true curve belongs, an appropriate choice of the penalty P will help in reducing the bias of the estimate, thus giving a better estimate.

2.2.3 Some Theory on Lsplines

Let us introduce the motivation behind the choice of a penalty along with some theory.

We now know that through its kernel, the penalty is associated with what we called a favoured parametric model. We will consider penalties that are based on differential operators. First, let us assume that the regression function m is in $\mathcal{H}^k$, the Sobolev space of functions with $k-1$ absolutely continuous derivatives and square-integrable k^{th} derivative. We will consider the penalized least squares problem of the form (2.32) where $\hat{m}$ is the minimizer.

Specifically, let L be a k^{th} order linear differential operator

$$L = D^k + \sum_{i=0}^{k-1} w_i D^i \tag{2.35}$$

where D^i represents the i^{th} derivative operator and the w_i's are continuous real-valued weight functions. In (2.32) we take the penalty to be

$$P(m) = \int (Lm)^2 \tag{2.36}$$

and we want to find m that minimizes the following

$$C(m, \mathbf{w}) = \sum (Y_i - m(x_i))^2 + \lambda \int [(D^k + \sum_{i=0}^{k-1} w_i D^i)m]^2, \tag{2.37}$$

where $\mathbf{w} = (w_0, \ldots, w_{k-1})'$. We call the minimizer an Lspline.

Choosing a penalty L is equivalent to choosing a favoured parametric model, i.e., the set of functions with $Lm = 0$. Here are a few examples of differential operators and their respective bases for the corresponding parametric families.

Table 2.3: *Differential Operators and their Parametric Family*

Operator L	Parametric Family for kernel of L
D^2	$\{1, t\}$
$D^2 + \gamma D$, $\gamma \neq 0$	$\{1, \exp(-\gamma t)\}$
D^4	$\{1, t, t^2, t^3\}$
$D^3 + \omega^2 D$, $\omega \neq 0$	$\{1, cos\, \omega t, sin\, \omega t\}$
$D^4 + \omega^2 D^2$, $\omega \neq 0$	$\{1, t, cos\, \omega t, sin\, \omega t\}$
$(D^2 - \gamma D)(D^2 + \omega^2 D)$ $\gamma \neq 0$, $\omega \neq 0$	$\{1, \exp \gamma t, cos\, \omega t, sin\, \omega t\}$

Given L, one can easily find the corresponding basis functions in the case where the w_i's are assumed to be constant real numbers. For that case, the following Theorem gives a simple recipe for calculating the basis functions for the space of m with $Lm = 0$. For a proof of this result, the reader is referred to the standard differential equation results of Coddington (1961).

Theorem 2 *Let us denote the* p *distinct roots of the polynomial* $x^k + \sum_{i=0}^{k-1} w_i x^i$ *as* $r_1, \ldots, r_p$ *and let* m_j *denote the multiplicity of root* r_j. *Then the following functions of* t *form a basis for the kernel of the differential operator* L*:*

$$\exp(r_j t), t\exp(r_j t), \ldots, t^{m_j - 1}\exp(r_j t) \qquad j = 1, \ldots, p.$$

The other way to proceed is to specify a desired basis and then find the corresponding penalty L. For certain favoured parametric models, the differential operator L is pretty easy to calculate. Let us assume that the favoured parametric model is a linear combination of $\mu_0, \ldots, \mu_{k-1} \in \mathcal{H}^k$ which are linearly independent with each having k continuous derivatives. Since there are k functions in the basis, an L of order k can be found, if the associated Wronskian matrix $[\mathbf{W}(t)]_{ij} = \mu_{i+1}^{(j-1)}(t)$ is invertible for all $t \in [a, b]$. One can easily find the weight functions $w_0, \ldots, w_{k-1}$ by solving

$$0 = (L\mu_i)(t) = \mu_i^{(k)}(t) + \sum_{j=0}^{k-1} w_j(t)\mu_i^{(j)}(t),$$

which is equivalent to

$$\mu_i^{(k)}(t) = -\sum_{j=0}^{k-1} w_j(t)\mu_i^{(j)}(t).$$

This last expression can be written in the following matrix form:

$$\mathbf{W}(t)\begin{pmatrix} w_0(t) \\ \vdots \\ w_{k-1}(t) \end{pmatrix} = -\begin{pmatrix} \mu_0^{(k)}(t) \\ \vdots \\ \mu_{k-1}^{(k)}(t) \end{pmatrix} \tag{2.38}$$

that is

$$\begin{pmatrix} w_0(t) \\ \vdots \\ w_{k-1}(t) \end{pmatrix} = -\mathbf{W}(t)^{-1} \begin{pmatrix} \mu_0^{(k)}(t) \\ \vdots \\ \mu_{k-1}^{(k)}(t) \end{pmatrix}. \tag{2.39}$$

Data analysts usually choose models by using the knowledge that they have of the data to be analysed. Note that in nonlinear regression, many of the parametric models are based on physical models, so the favoured parametric model could also be chosen that way. In our case for example, since ozone levels are known to have a periodic pattern, we would like to use a $\{1, sin, cos\}$ model to estimate m. Using this basis made of three functions, it would then be easy, using the previous theory, to find a penalty L of order 3.

In addition, it is important to consider the goal of the analyses: do we want to estimate the regression function m or one of its derivatives? In the first case, a differential operator of order 2 or greater is suggested while in the latter case, an operator of order 4 or higher would perform better.

It is also possible to define the effective number of parameters used (p_λ) by a smoothing spline. Here we will use the definition given by Hastie and Tibshirani (1990)

$$p_\lambda = trace\ \mathbf{S}_{\lambda,\mathbf{w}} \tag{2.40}$$

where $\mathbf{S}_{\lambda,\mathbf{w}}$ is a hat matrix, defined as

$$S_{ij}(\lambda) = \frac{1}{n} G(x_i, x_j) \tag{2.41}$$

with G the weight function from (2.33). It can be shwon that $\mathbf{S}_{\lambda,\mathbf{w}}$ is independent of $\mathbf{Y} = (Y_1, \ldots, Y_n)$, and satisfies the following $\hat{\mathbf{m}}_\lambda = \mathbf{S}_{\lambda,\mathbf{w}}\mathbf{Y}$ for

$\hat{\mathbf{m}} = (\hat{m}_\lambda(x_1), \ldots, \hat{m}_\lambda(x_n))'$. One can show that, for fixed $\mathbf{w}$, p_λ is a decreasing function of λ with $p_\lambda = n$ when $\lambda = 0$ and $\lim_{\lambda\to\infty} p_\lambda = n_L$, where n_L is the dimension of the kernel of L. For example, in the case where we use the favoured parametric model $\{1, sin, cos\}$, we get $n_L = 3$. Thus, a small smoothing parameter is equivalent to a large number of parameters and it results in a "wiggly" estimate of m while a large smoothig parameter (a small number of parameters) yields a smoother estimate of the regression function. Note that if we choose a data-dependent smoothing parameter $\hat{\lambda}$, we then have $\hat{\mathbf{m}}_{\hat{\lambda}} = \mathbf{S}_{\hat{\lambda},\mathbf{w}}\mathbf{Y}$ where $\mathbf{S}_{\hat{\lambda},\mathbf{w}}$ now depends on $\mathbf{Y}$ via $\hat{\lambda}$. Nonetheless, the trace of $\mathbf{S}_{\hat{\lambda},\mathbf{w}}$ is still used to define p_λ. In the case where $\mathbf{w}$ is estimated, the number of parameters used by an estimate is given by $trace\ \mathbf{S}_{\hat{\lambda},\hat{\mathbf{w}}}$ + length $\mathbf{w}$.

For more details on the theoretical aspects of smoothing splines with model based penalties, please refer to Heckman and Ramsay (1996). The Splus module *Lspline*, written by Ramsay and available on the Statlib web site (http://lib.stat.cmu.edu), allows the data analyst to compute splines with model based penalties easily and quickly.

Later on in this thesis, when we reach the segment of data simulations and data analysis, we will use the method of splines with model-based penalty. The choice of the penalty along with the motivations will be discussed in Chapter 4. However, before doing so, we will introduce in the next section the different automated techniques of smoothing parameter selection in the case of spline smoothing when the observations are independent. In Chapter 3, we will introduce techniques that are specifically designed for correlated regression errors, the subject of this thesis. In Chapters 4 and 5, we will apply two of these techniques to a simulation study and to a real data set.

2.2.4 Smoothing Parameter Selection for Smoothing Splines

Spline smoothing has become a very popular method for estimating the regression function in a nonparametric model. As discussed earlier, the performance of the spline smoothing technique depends on the choice of the penalty but most importantly on the choice of the smoothing parameter. There exist many methods of selecting the smoothing parameter when the observations are assumed to be independent. Such methods include cross-validation (CV) and generalized cross-validation (GCV) among the most commmonly used.

We will now introduce the CV and GCV methods in the context of data that are assumed to be independent and then, in Chapter 3, will present a few methods for the case of correlated data.

Cross-Validation

We saw in section 2.1.4 the idea of cross-validation for polynomial regression. It is basically the same idea that is used for spline smoothing. Let $\hat{m}_\lambda^{(i)}$ be the smoothing spline obtained from all the data except Y_i using a smoothing parameter value of λ. Then, the cross-validation choice of λ is the value of λ that minimizes the CV criterion

$$CV(\lambda) = \frac{1}{n}\sum_{i=1}^{n}\left(Y_i - \hat{m}_\lambda^{(i)}(x_i)\right)^2. \tag{2.42}$$

Now recall the hat matrix defined in equation (2.41). Craven and Wahba (1979) showed that equation (2.42) has an easier computational form:

$$CV(\lambda) = \frac{1}{n} \sum_{i=1}^{n} \frac{(Y_i - \hat{m}_\lambda(x_i))^2}{(1 - S_{ii}(\lambda))^2}, \tag{2.43}$$

where $\hat{m}_\lambda$ is the smoothing spline obtained using all data points.

Generalized Cross-Validation

Craven and Wahba (1979) also suggested another criterion which is based on CV. It is called generalized cross-validation (GCV) and is obtained by replacing $S_{ii}(\lambda)$ in (2.43) by its average value, $n^{-1}trS(\lambda)$. We then obtain the GCV criterion to be minimized

$$GCV(\lambda) = \frac{1}{n} \frac{\sum(Y_i - \hat{m}_\lambda(x_i))^2}{\{1 - n^{-1}trS(\lambda)\}^2}, \tag{2.44}$$

Theoretical arguments show that GCV should, asymptotically, give the best choice possible for the value of the smoothing parameter λ in the sense of minimizing the average squared error at the design points. For more details on the computational aspect of GCV, please refer to Silverman (1984b).