ECO 523 - Class 3

Nonparametric Econometrics

Carolina Caetano*

Contents

1 Series based methods

Kernel methods estimate the values of the function $g(x)$ for each x individually. They do that by taking in consideration only the observations closest to x. Hence, any abnormality of the function that happens away from x is of no consequence for kernel estimation, provided the sample is large enough. Because of this, kernel methods are the core nonparametric methods.

There are three main disadvantages of kernel methods. The first is computational. Estimating $g(x)$ requires only one regression, but if the researcher needs to estimate $g(x)$ for many values of x, each of them requires a new regression, and this can add up quickly. This is very often the case when the nonparametric estimation is a plugin inside of a more complex estimator. Such cases usually require the estimation of $g(X_i)$ for all X_i in the sample, and this will therefore require somewhere around n regressions. The estimation of the variance of $\hat{g}(x)$, for example, is one such case.

Another disadvantage of the kernel methods is the difficulty of including information or restrictions over the functional form into the estimation process. Often, restrictions can be

*Special thanks to Yusuke Jinnai for help with plots and editing.

incorporated into the estimation process with the local polynomial, but only for efficiency purposes. The resulting estimated function $\hat{g}$ may actually not satisfy the restriction itself.

The third disadvantage of kernel methods is a tendency to multicollinearity. For example, if we are using local polynomial, there is the added difficulty that higher terms of the expansion tend to become multicollinear if the bandwidth is small. The idea is that if $X_i - x$ is close to zero, then $(X_i - x)^3$ and $(X_i - x)^4$ are in fact quite similar. Even if we use simple kernel regression, there is still a serious tendency to multicollinearity in the multivariate cases. The idea is that there must exist observations X_i that are within boundary distance of x. However, depending on x, this may be impossible in practice. For example, if x represents a very low income level and a very high education level, there may not be many observations within bandwidth range of this point.

Series methods solve these problems. One single regression is capable of yielding estimates of $g(x)$ for all x in its domain. Also, restrictions can be incorporated automatically for efficiency gains, but more than that, the resulting $\hat{g}$ will satisfy the restrictions. The biggest weakness of series methods is exactly in its strength: the method bunches so much information into a single regression that function irregularities are often not perceived.

1.1 The orthogonal regression

Let the model be, as usual,

$$Y = g(X) + u$$

where $g(X) = E(Y|X)$. To begin, suppose that X is scalar and $g \in \mathcal{C}^\infty$, then, by the Taylor expansion,

$$g(X_i) = \sum_{k=0}^{\infty} \frac{g^{(k)}(0)}{k!} X_i^k.$$

Consider the functions:

$$p_0(x) = 1, \quad p_1(x) = x, \quad p_2(x) = x^2, \quad p_3(x) = x^3, \quad p_4(x) = x^4, \quad p_5(x) = x^5, \quad \dots$$

then

$$g(X_i) = \sum_{k=0}^{\infty} \beta_k p_k(X_i).$$

If $\mathcal{C}^\infty$ is seen as an infinite-dimensional vector space, then $\{p_0, p_1, p_2, p_3, p_4, p_5, \dots\}$ is a basis of $\mathcal{C}^\infty$.

We will try to approximate $g(X_i)$ by projecting the Y_i into the $K+1$ dimensional space

spanned by $\{p_0, p_1, \ldots, p_K\}$. The way to do this is by OLS regression:

$$\min_{\beta_0,\beta_1,\ldots,\beta_K} \sum_{i=1}^{n} (Y_i - \beta_0 p_0(X_i) - \beta_1 p_1(X_i) - \cdots - \beta_K p_K(X_i))^2.$$

Hence, in matrix notation, if

$$\beta = \begin{bmatrix} \beta_0 \\ \vdots \\ \beta_K \end{bmatrix}, \; Y = \begin{bmatrix} Y_1 \\ \vdots \\ Y_n \end{bmatrix}, P = \begin{bmatrix} p_0(X_1) & p_1(X_1) & \ldots & p_K(X_1) \\ p_0(X_2) & p_1(X_2) & \ldots & p_K(X_2) \\ \vdots & \vdots & \vdots & \vdots \\ p_0(X_n) & p_1(X_n) & \ldots & p_K(X_n) \end{bmatrix},$$

then

$$\hat{\beta} = (P^T P)^{-1} P^T Y,$$

and therefore:

$$\hat{g}(x) = \sum_{k=0}^{K} \hat{\beta}_k p_k(x).$$

As you can see, the estimator requires only one regression for the estimation of β, and from there $\hat{g}(x)$ is simply a matter of calculating the value of the function above for each particular x, which computers can do very quickly.

Hence, in the Taylor expansion example, the estimated function would be

$$\hat{g}(x) = \hat{\beta}_0 + \hat{\beta}_1 x + \hat{\beta}_2 x^2 + \cdots + \hat{\beta}_K x^K,$$

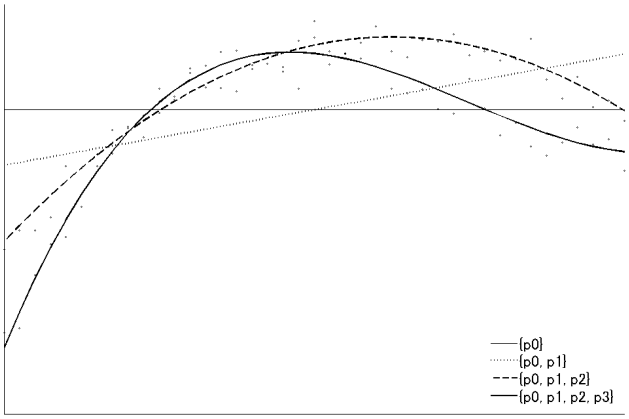
and Figure 1 shows $\hat{g}(x)$ with the basis of $\{p_0\}$, $\{p_0, p_1\}$, $\{p_0, p_1, p_2\}$, $\{p_0, p_1, p_2, p_3\}$, respectively.

A multivariate power basis can be achieved by combining the powers of each of the coordinates. For example, the bivariate power basis is:

$$\{1, \; x_1, \; x_2, \; x_1 x_2, \; x_1^2, \; x_2^2, \; x_1 x_2^2, \; x_1^2 x_2, \; x_1^2 x_2^2, \; x_1^3, \; \ldots\}.$$

Observe that the OLS regression is equivalent to a very small multivariate power series expansion.

Figure 1: $\hat{g}(x)$ with several power bases



1.1.1 The importance of orthogonality and how to program $\hat{g}$

If the basis is orthonormal, then $(P^T P) = I_K$. This is a very desirable computational property for a basis. If this is the case,

$$\hat{\beta} = P^T Y = \sum_{i=1}^{n} \begin{bmatrix} p_0(X_i) \\ \vdots \\ p_K(X_i) \end{bmatrix} Y_i$$

and therefore:

$$\begin{aligned} \hat{g}(x) &= \sum_{k=0}^{K} p_k(x) \sum_{i=1}^{n} p_k(X_i) Y_i \\ &= \sum_{i=1}^{n} \left[\sum_{k=0}^{K} p_k(x) p_k(X_i) \right] Y_i \end{aligned}$$

which is very easy to compute.

The bases can be orthonormalized. For example, suppose that the domain of the function of the interval $[0, 1]$, and the space is L^2. Then the power basis can begin with $p_0(x) = 1$, then $p_1(x) = \frac{x - \frac{1}{2}}{\sqrt{\int_0^1 \left(x - \frac{1}{2}\right)^2 dx}} = \sqrt{3}(2x - 1)$, then $p_2(x)$ is the degree 2 polynomial which is orthogonal to both p_0 and p_1, normalized: $p_2(x) = \frac{x^2 - x + \frac{1}{6}}{\sqrt{\int_0^1 \left(x^2 - x + \frac{1}{6}\right)^2 dx}} = \sqrt{5}(6x^2 - 6x + 1)$, and so on..

Hence, if you are using an arbitrary basis, the series regression can be programed as any regression. If (more likely) you are using a well known orthonormal basis, then

you can program the series estimator as simple sums. In a matrix-based software, let $P(x) = (p_0(x), p_1(x), \ldots, p_K(x))$, then

$$\hat{g}(x) = P(x)P^T Y.$$

1.2 Heteroskedasticity

In the kernel regression cases, the estimation of $g(x)$ was done per point, and therefore there was no concern about heteroskedasticity.[1] In the series regression cases, since the method is global, the simple least squares method adopted may not be the most efficient if the u's are heteroskedastic.

If you have a model of the behavior of the u, you can easily incorporate it to the regression and perform a FGLS instead of the simple OLS procedure we have been advocating. Proceed exactly as you would do the parametric case, except that instead of the regression being onto the X_i, it is into the values $p_0(X_i), p_1(X_i), \ldots, p_K(X_i)$.

In the remainder of this and the next class we will still use the simple least squares. When we work on the asymptotic results on the next class we will allow for heteroskedasticity in the calculations of the variance.

1.3 Why the method works

Observe that

$$\begin{aligned}
|\hat{g}(x) - g(x)| &\leqslant \left|\hat{g}(x) - \sum_{k=0}^{K} \beta_k p_k(x)\right| + \left|\sum_{k=0}^{K} \beta_k p_k(x) - g(x)\right| \\
&= \left|\sum_{k=0}^{K} \hat{\beta}_k p_k(x) - \sum_{k=0}^{K} \beta_k p_k(x)\right| + \left|\sum_{k=0}^{K} \beta_k p_k(x) - \sum_{k=0}^{\infty} \beta_k p_k(x)\right| \\
&= \left|\sum_{k=0}^{K} \left[\hat{\beta}_k - \beta_k\right] p_k(x)\right| + \left|\sum_{k=K+1}^{\infty} \beta_k p_k(x)\right| \\
&\leqslant ||\hat{\beta} - \beta|| \cdot ||P(x)|| + \left|\sum_{k=K+1}^{\infty} \beta_k p_k(x)\right|
\end{aligned}$$

for the Euclidean norm.

This kind of decomposition is at the root of the reason why these approximations work. The second part depends entirely on the ability to express g in terms of $\{p_0, p_1, p_2, p_3, p_4, p_5, \ldots\}$ (that is, g belongs to the vector space spanned by this basis), and that the basis be good enough that the approximation with the first $K+1$ elements is indeed a good

[1] It is possible to improve kernel based estimation in small samples if one takes heteroskedasticity in consideration.

approximation. The larger K, the more elements in the approximation, and therefore the smaller this term. Notice also that this term is deterministic, and it has a direct correspondence to the bias of the estimation. Hence, the larger K, the smaller the bias.

The term $||P(x)|| = \left[\sum_{k=0}^{K} p_k(x)^2\right]^{1/2}$ is assumed to increase slowly as $n \to \infty$, and this is not an issue for the bases that we commonly use. Finally, the only random term is $||\hat{\beta} - \beta||$, which determines, therefore, the variance of the estimator. The larger K, the higher the variance of $\hat{g}$. This is easy to see: the larger K, the more coefficients to estimate. It is a well known fact of the OLS regression that, for a given sample size, the larger the number of explanatory variables, the higher the variance of the estimation of each coefficient. This happens because the closer to multicollinearity, the higher the variance of the OLS, and the inclusion of one new covariate necessarily approximates the covariates to perfect multicollinearity, unless it is entirely uncorrelated with the rest.

Therefore, the choice of the size of the basis to use is the same kind of issue as the choice of the bandwidth in kernel regression. The more elements are used, the smaller the bias, and the larger the variance. The choice of basis is theoretically similar to the choice of kernel, but in practice it is a lot more important.

1.4 Choice of basis

The goal is to choose the basis that is capable of better approximating the function with the least elements. This way we can decrease the bias without affecting the variance of the estimation. In general, this depends on the information that we have about the function. For example, if we know that the function is increasing, we may use a basis that is entirely increasing.

The polynomial basis, though one of the first to be used and studied, is generally considered a poor basis. The first problem is known as the Runge's phenomenon: since the power polynomials are forced to vary somewhere, usually as the number of elements increase this variation is pushed to the boundary. Consequently, a great part of the function is poorly approximated, and the problem can get worse the more elements of the series (higher power polynomials) are being used. Second, because the polynomial basis is very global, it does a poor job at any particular point, and it is extremely sensitive to outliers.

1.4.1 The Fourier basis

The Fourier basis is one of the oldest, and though still sensitive to outliers and badly behaved at the boundary, it is generally a better basis than the power series. It is excellent for approximating periodic functions, but can approximate other functions as well. If

$g \in L^2$, the Fourier expansion of the function g at all points where g is differentiable is:

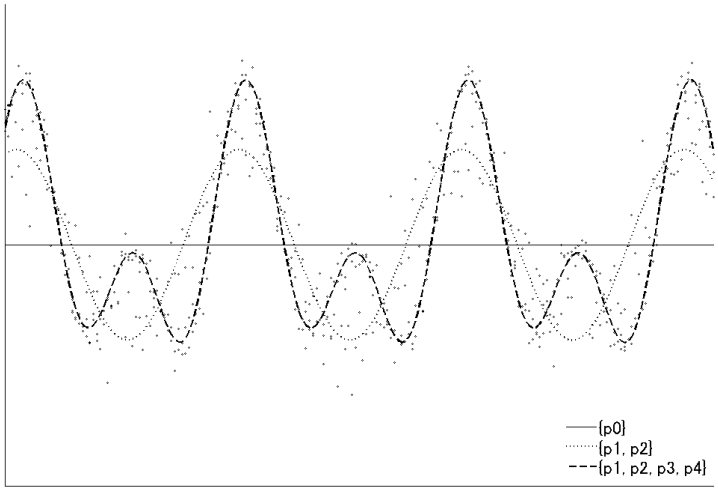
$$g(x) = \frac{a_0}{2} + \sum_{k=1}^{+\infty} a_k \cos(kx) + b_k \sin(kx),$$

where

$$a_k = \frac{1}{\pi} \int_{-\pi}^{+\pi} g(x) \cos(kx) dx$$
$$b_k = \frac{1}{\pi} \int_{-\pi}^{+\pi} g(x) \sin(kx) dx.$$

If $p_0(x) = 1$, $p_1(x) = \cos(x)$, $p_2(x) = \sin(x)$, $p_3(x) = \cos(2x)$, $p_4(x) = \sin(2x)$, $p_5(x) = \cos(3x)$, $p_6(x) = \sin(3x)$ etc., then $\{p_0, p_1, p_2, p_3, p_4, p_5, \dots\}$ is a basis of L^2. Figure 2 shows $\hat{g}(x)$ with the Fourier basis of $\{p_0\}$, $\{p_1, p_2\}$, and $\{p_3, p_4\}$, respectively.

Figure 2: $\hat{g}(x)$ with several Fourier bases



In practice, this expansion is only good if the domain of g is compact, and we are estimating only interior points. The fact is that the Fourier series is badly behaved at boundary points, and therefore does a poor job of approximating discontinuous functions. The reason is that the more elements of the series are used, the better the basis approximates the function all around. However, the necessary variation of the lower period elements has to be used somewhere, and it is generally thrown to the boundary, which is fitted in an increasingly worse manner the more elements are used. This is known as the Gibbs' phenomenon, and is of similar nature to Runge's phenomenon of power bases. The global

nature of the Fourier series makes it still sensitive to outliers, though in practice less so than power series.

Other bases that you may check out: Hermite polynomials, Laguerre polynomials, Legendre polynomials, Shifted Legendre polynomials, Chebyshev polynomials, Shifted Chebyshev polynomials, among others. In this class we will focus on two excellent bases: B-splines, and wavelets.

1.4.2 Spline bases

Spline bases are in general a better choice than most other bases. If you have to blindly choose a basis for a given problem, you should definitely choose a spline basis (unless you know that the function is discontinuous, has kinks or some other highly localized abnormality). Splines are only locally sensitive to outliers and have much better boundary behavior than power and Fourier series. This basis does a considerably better job of approximating functions with less elements, and has become the standard of series regression.

Suppose that the domain of the function g is compact, and that we divide it in K subintervals. Let t_0 be the left boundary, $t_1, t_2, t_3, \ldots, t_{K-1}$ be the interval dividing points and t_K be the right boundary. The t's are called the **knots** of the spline function and $t = (t_0, t_1, \ldots, t_K)$ is called a **knot vector**. If the t's are equidistant, the spline is called **uniform**.

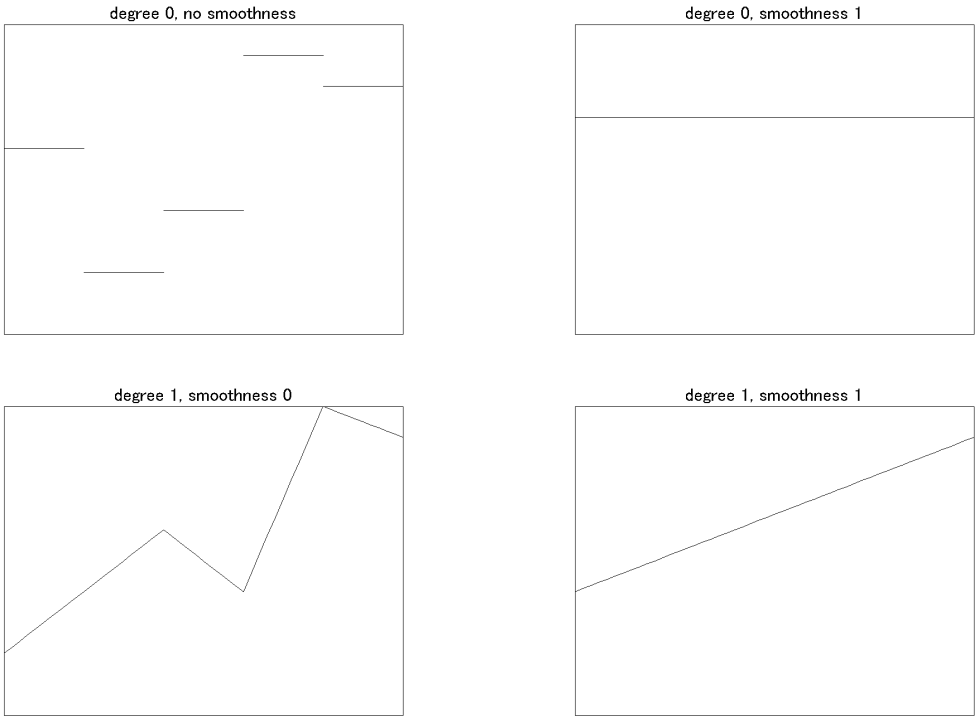
A spline function of **degree** r (or **order** $r+1$) and **smoothness** m is a piecewise polynomial function such that at each of the subintervals it is a polynomial of degree r, with the restriction that at the knots the polynomials must connect, and more than that, their derivatives up to the m^{th} degree have to match. The resulting function (spline) is a $\mathcal{C}^m$ function, and the loss of smoothness is therefore $r - m$ at the most.

Figure 3 shows splines with several degrees and smoothness. A degree 0 spline is a step function. A degree 1 spline with smoothness 0 is a piecewise linear function. A degree 1 spline with smoothness 1 is a straight line. The most traditional of all the splines has degree 3 and smoothness 2. This is a piecewise cubic polynomial that belongs to $\mathcal{C}^2$, because the cubic polynomials are twice continuously differentiable everywhere inside the intervals, and at the knots as well because of the smoothness level 2. When the second derivatives of the spline at t_0 and at t_K are set to zero (so that t_0 and t_K are inflection points), this spline is called a **natural** spline.

The space of all the spline functions of degree r, smoothness m and knots $t = (t_0, \ldots, t_K)$ is a vector space (where the vectors are piecewise cubic polynomial functions), and it is commonly denoted as $S_r^m(t)$.

Originally, spline functions were used to interpolate data. With noisy data, as is the case in this class, spline smoothing is the regression technique resulting from the minimization

Figure 3: Splines with several degrees and smoothness



problem

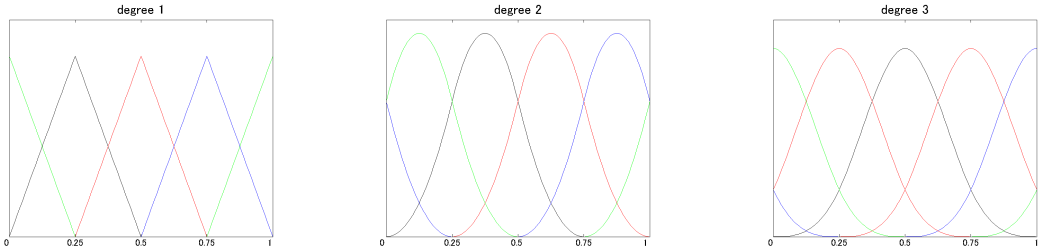
$$\min_{s(x)\in S_r^m(t)} \frac{1}{n}\sum_{i=1}^{n}(Y_i - s(X_i))^2 + \lambda \int_a^b (s^{(m)}(x))^2 dx$$

for some $\lambda > 0$. The higher λ, the harder the punishment for non-smoothness. For example, if $m = 2$, λ is punishing curve sinuosity. A straighter curve will not be punished as much, but will probably not fit the data as well. The technique above is known as **spline smoothing**. It is extremely related to spline basis, but we will not study it.

The technique we will study is series regression with a spline basis. This basis arises from the following result: any spline of degree r, smoothness m and knot vector t can be represented as a linear combination of a particular group of splines of the same degree, smoothness and knot vector. This group is known as the **B-spline** basis. A B-spline basis size is guided by the number of intervals. It is a set of $K + r$ spline functions of degree r and knot vector t (the "$+r$" will be explained below). Observe that form most softwares the smoothness level is implicitly defined in the knot vector. If no knots are repeated, the smoothness is the maximum level $r - 1$. Each time a knot is repeated, one degree of smoothness is lost on that knot.

When the knots are equally spaced (uniform splines), the r degree B-spline basis functions with knot vector t are simple horizontal shifts of each other. Suppose that the

Figure 4: Uniform B-spline basis with several degrees



support of the distribution of X is $[0,1]$,[2], then each element in the B-spline basis has a simple formula:

$$B_r(x|t) = \frac{1}{a^r r!} \sum_{j=0}^{r+1} (-1)^j \binom{\text{r+1}}{\text{j}} [\max\{0, x - t_j\}]^r$$

It's easy to see that $B_r(x|t) = 0$ for $x < t_0$ and $x > t_K$. The B-spline basis is therefore $p_1(x) = B_r(x|t)$, $p_2(x) = B_r\left(x - \frac{1}{K}\middle|t\right)$, $\ldots$, $p_K(x) = B_r\left(x - \frac{K-1}{K}\middle|t\right)$. However, this basis will necessarily estimate $g(t_0) = g(t_K) = 0$. To avoid this, the B-spline basis incorporates more elements, using all the dislocations of size $1/K$ below t_0 and above t_K that yield a non-zero function in $[0,1]$. For example, if $r = 1$, then we add one element to the basis: $p_{K+1}(x) = B_r\left(x + \frac{1}{K}\middle|t\right)$. In general, a uniform B-spline basis will have $K + r$ elements. Figure 4 shows uniform B-spline basis with several degrees when $K = 4$. The extra elements are equivalent to the extension of the knot list, and therefore, the new list is known as the **extended knot vector**.

Observe that this is an orthonormal basis, and therefore it can be easily programed. In Stata, look for the command bspline. This package does the spline regression automatically. The option "power" is the degree of the spline, the knots give the subintervals, but remember they also give the smoothness level if you repeat the internal knot values. The package extends the knot list for you, unless you specify otherwise. I believe that the package only works for scalar X, but I am not sure. If this is the case, it's an unfortunate caveat, but there are workarounds. See about multivariate splines below.

It is a result of approximation theory that when the interval size decreases (and the number of intervals K increases), the spline basis can approximate functions in L^2. Hence, B-spline bases can be used as bases in nonparametric series regressions.

In practice, since the support of the B-spline basis functions shifts, each of the basis functions is acting in one region of the domain of g more or less independently of the other basis functions. Because of this feature, the spline basis can pick up local behavior, while still being a global method. This approach has a resulting $\hat{g}$ which is much more sensitive

[2]This is without loss of generality if the support of the distribution of X is compact and known.

to local variation than global approaches such as power and Fourier series. It is possible to achieve better global fits with splines, even though the degree of the polynomial remains low. Since the polynomial degree remains fixed (and hardly ever above 3), boundary issues of the Runge's and Gibb's phenomenons kind are entirely avoided.

The intuition of the method is the following: imagine that an OLS regression with a polynomial of degree r is being performed in each subinterval, with the restriction that the resulting fitted curves have to match at the knots. Spline basis are not that local, but are not too distant from that either.

From this kind of reasoning it is very easy to see what drives the bias and variance of the method. If K is large, and the intervals are small, the estimation is using more elements of the basis, which usually decreases bias and increases variance. However, in the case of splines it is easy to see why. The smaller intervals imply that the basis elements are fitting a group of data that is closer together, and therefore the bias of that fit will be smaller. At the same time, the smaller intervals imply that less data is being fitted by each basis function, and therefore the variance increases.

Even though splines behave much better for boundary estimation than global methods, it is not the ideal method for boundary estimation (at t_0 or t_K). To see why, suppose we are estimating $g(t_0)$. The value of the function at t_0 is being determined by basis functions that use mostly observations such that $t_0 \leqslant X_i < t_1$. However, the level and derivative matching is strongly forcing the behavior of the function at t_1, while there is no constraint on t_0's side. The result is that there will still be large bias at t_0.

The variables of choice for spline bases are the knots $t_1, t_2, t_3, \ldots, t_{K-1}$ and the degree of the polynomial. For finite samples, the degree of the polynomial has a similar effect to the number of intervals. The higher the polynomial degree, the better the fit, but the bigger the variance.

A real issue with splines is overfitting. If the intervals are small enough and the polynomial degree is sufficiently large, it is sometimes possible to fit almost all of the data perfectly. Hence, even though the MSE will be very small, the predicted $\hat{g}$ is not really close to g. This is natural of the method. Splines flourished first as an interpolation method, and as that they are excellent. Hence, when using splines to deal with noisy data, it is advisable to maintain the degree of the polynomial relatively low, so as to avoid this problem, and also Gibbs' boundary issues, and modify the interval sizes as an adjustment method. Usually a cubic spline with smoothness degree 2 is widely accepted, and it is common to perform robustness checks with degree 1 and 2 polynomials.

The question of the choice of the size (and therefore number) of intervals is analogous to the choice of the number of elements in any other orthogonal basis, since the bias-variance trade-off is the same.

Multivariate splines: For general multivariate regression models, $X \in \mathbb{R}^q$, the product of spline functions are also spline functions themselves. They are known as **tensor splines**, and to form a basis it is enough to cross-multiply the univariate corresponding basis elements.

This is only possible in practice if the multivariate function's support is a rectangle. This can be arranged by dropping part of the sample. If this cannot be done, as is the case if the covariates are highly correlated and form a pattern that cannot be cut into a rectangle without serious loss of data, then it is advisable to use a general multivariate spline base function to approximate $g(x)$. Chui (1988) and Eubank (1999) wrote on the construction of multivariate spline bases.