Statistical Modeling with Spline Functions
Methodology and Theory

Mark H. Hansen
University of California at Los Angeles

Jianhua Z. Huang
University of Pennsylvania

Charles Kooperberg
Fred Hutchinson Cancer Research Center

Charles J. Stone
University of California at Berkeley

Young K. Truong
University of North Carolina at Chapel Hill

January 5, 2006

6
Density Estimation

6.1 An example

6.1.1 The income data

In Figure 6.1 we show a histogram of a random sample of 7,125 annual net incomes in the United Kingdom [Family Expenditure Survey (1983); the data have been rescaled to have mean one as in Wand, Marron, and Ruppert (1991).] There is a large group of people having almost identical incomes of about 0.24, due to the national UK old age pension. Depending on how many bins we choose, this peak can be quite a bit sharper than that in Figure 6.1. The peak is also somewhat recognizable in the subplot on the left side of Figure 6.2, which zooms in from the quantile-quantile (QQ) plot against a normal distribution, keeping the aspect ratio the same as in the complete plot. We note that when the income is about 0.24 the quantile-quantile plot is quite flat. It is fairly hard to recognize the peak from either of these two plots, as the removal of approximately 100–150 “extra” people between 0.22 and 0.27, or only 2% of the total sample, would remove the peak.

The plot on the right side of Figure 6.2 is a quantile-quantile plot for the income data on a log-scale. The line in this plot corresponds to a log-normal distribution with the same 80th and 90th percentile as the income data. From this plot we note that the income density has a heavier right tail than a log-normal distribution.

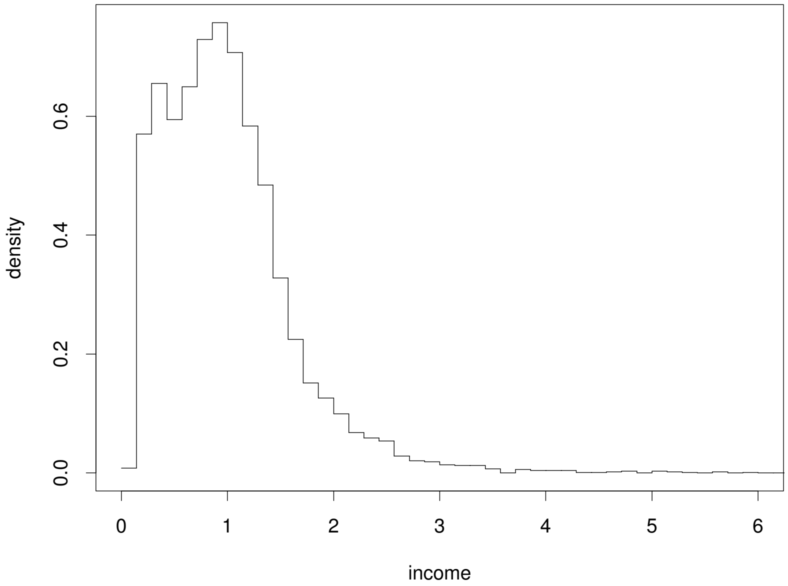

FIGURE 6.1. Histogram of the income data.

6.1.2 Background

Suppose that we are interested in estimating the density of incomes in the UK. This means that we want to identify a density function f so that the income values $Y_1, \ldots, Y_{7125}$ are iid according to f. The easiest way to identify f is to assume that the density comes from a gamma, Weibull, lognormal, or other classical parametric family. However, we know that for the income data this will not give a satisfactory result: all of these families will fail to model the peak, and they will have trouble accurately modeling the heavy tail.

While it is trivial to estimate the distribution function F nonparametrically by the empirical distribution function, this does not yield a useful estimate of the density function. The "derivative" of the empirical distribution function is a collection of point masses at the data points.

We can think of a histogram as a density estimate that models a density function as a piecewise constant function and estimates the unknown coefficients of the model by the method of maximum likelihood. Formally we can define a histogram density estimate f_{hist} as follows: given an origin a and a binwidth h, set

$$\widehat{f}_{\text{hist}}(y) = \frac{1}{n}\sum_{i=1}^{n} I(jh + a \leq Y_i < (j+1)h + a), \qquad j = \lfloor (y-a)/h \rfloor.$$

The histogram in Figure 6.1 is $\widehat{f}_{\text{hist}}$ with $a = 0$ and $h = 1/7$.

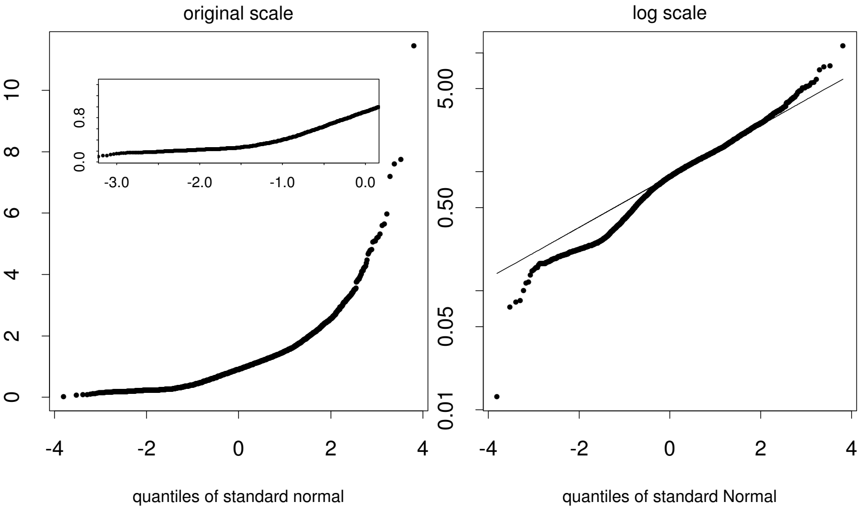

FIGURE 6.2. Quantile-quantile plots of the income data. The subplot on the left side maintains the same aspect ratio as the larger plot.

Probably the most commonly used method of density estimation is kernel density estimation. A kernel density estimate $\widehat{f}_{\text{kern}}$ with constant bandwidth can be defined as follows: given a nonnegative kernel function K with $\int K(x)\,dx = 1$ and given a positive bandwidth h, set

$$\widehat{f}_{\text{kern}}(y) = \frac{1}{nh}\sum_{i=1}^{n} K\Big(\frac{Y_i - y}{h}\Big), \qquad -\infty < y < \infty.$$

In Figure 6.3 we show a kernel estimate with the standard normal density as the kernel and $h = 0.4$ as the bandwidth.

In practice, the choice of the origin a for the histogram estimate and the choice of $K(\cdot)$ for the kernel estimate are not crucial. However, the choice of the bandwidth (binwidth) h for each method is of utmost importance. If h is chosen too small, the density estimate ends up being too spiky, while if h is too large, many details are smoothed away. For example, Figure 6.4 gives histogram and kernel density estimates with a bandwidth that gives a reasonable height for the peak (dotted lines) and with a bandwidth that gives a fairly smooth tail (solid lines). It is clear that with a fixed bandwidth, kernel and histogram density estimates cannot adequately describe this data. Still, kernel density estimation with a fixed bandwidth is exceedingly popular, and it is often the only density estimation method (other than histograms) available in statistical packages. Because of its conceptual simplicity, there has been an enormous amount of research on kernel density estimation, which has given rise to various rules for selecting an "optimal" bandwidth. In addition, various people have proposed transformations before a fixed bandwidth is chosen, as well as algorithms that let

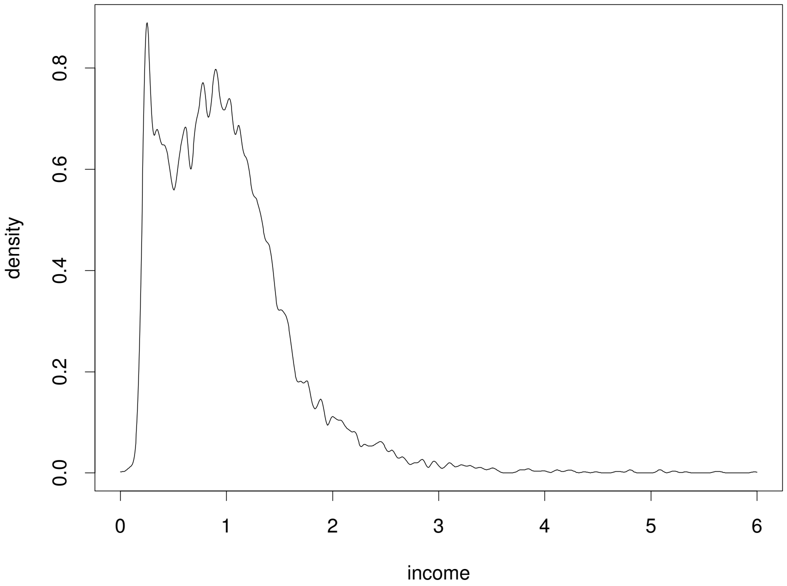

FIGURE 6.3. Kernel density estimate for the income data.

the bandwidth vary (i.e. a smaller bandwidth near the peak and a larger bandwidth in the tail).

There are many density estimation methods that give dramatically better estimates than histograms and kernel density estimates with a fixed bandwidth (for example, Logspline). However, all these methods share the problem that a *smoothing* parameter, like h, needs to be chosen.

6.1.3 Logspline density estimation

Motivated by the piecewise constant nature of the histogram density estimate, we can model the density function by a linear spline (continuous, piecewise linear function), quadratic spline (continuously differentiable, piecewise quadratic polynomial) or cubic spline (twice continuously differentiable, piecewise cubic polynomial). For the Logspline methodology we model the log-density function instead of the density function and we use the maximum likelihood method to estimate the unknown coefficients. For a fixed set of knots, this is a well-behaved estimation problem since the log-likelihood function is strictly concave. Stepwise addition and deletion of knots is applied to obtain a final set of knots.

On the left side of Figure 6.5 we show a Logspline density estimate for the income data discussed in Section 6.1.1. The right side of Figure 6.5 zooms in on the neighborhood of the peak near 0.24. The letters below the panels of Figure 6.5 indicate the starting knots (s), the knots that were in the largest model of the stepwise procedure (m), and the knots for the

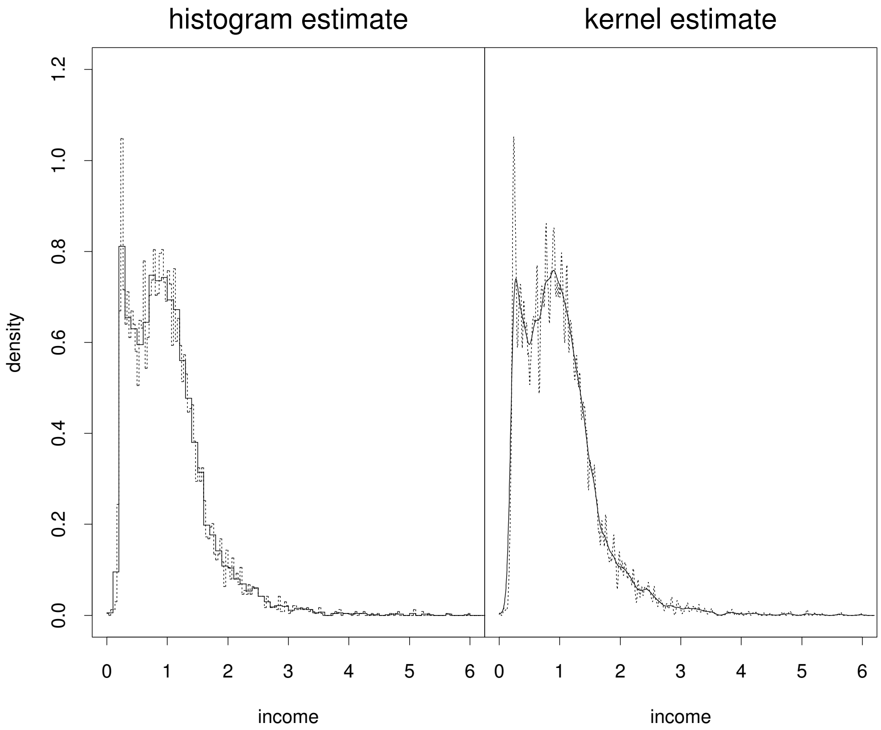

FIGURE 6.4. Histogram and kernel density estimates of the income data. For the solid curves the bandwidth was chosen to provide a smooth estimate of the tail; for the dotted curves the bandwidth was chosen to provide an accurate estimate of the height of the peak.

final model selected by BIC (f). Knot selection will be discussed in detail below. In Kooperberg and Stone (1992) we concluded that the height and the location of the peak are accurately estimated by Logspline.

In Figure 6.6 we show a probability-probability (PP) plot on a logit scale for the income data versus the Logspline fit to the density. Since this plot is virtually a straight line, we conclude that the Logspline estimate to the density for the income is quite good, giving tails of an appropriate heaviness.

There are some clear advantages to modeling the log-density function, rather than the density function. In particular:

- we do not need to be concerned about the requirement that a density be nonnegative: unconstrained optimization techniques will yield a density estimate that is positive; and
- the addition of a suitable constant to the estimate of the log-density will ensure that the density estimate integrates to one.

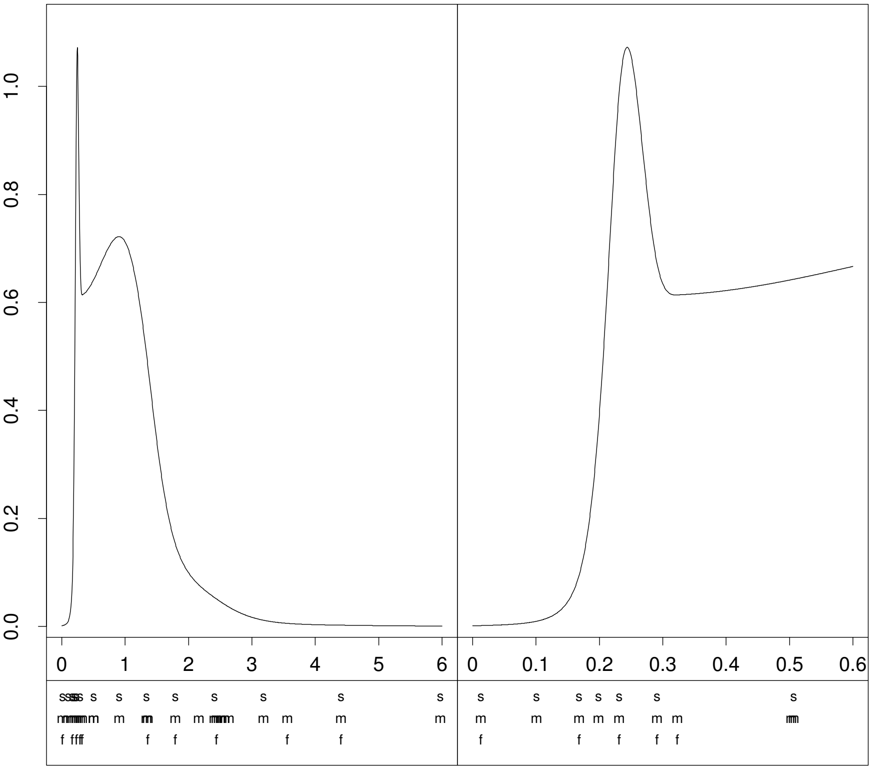

FIGURE 6.5. Left: Logspline density estimate for the income data; right: enlargement of the area near $x = 0.24$. The letters below the plots refer to the knot placement.

It is sometimes perceived as a disadvantage of modeling the log-density that the density estimate is strictly greater than zero, while the true density may equal zero for some values of y. If there is external information about the values of y such that the density equals zero, it is trivial to incorporate this information in the Logspline procedure. Otherwise, an estimate of, say, $\exp(-100) \approx 10^{-43}$ will usually be indistinguishable from zero for practical purposes.

6.2 The Logspline methodology

6.2.1 The Logspline model

Given the integer $K \geq 3$, the numbers L and U with $-\infty \leq L < U \leq \infty$, and the sequence $t_1, \ldots, t_K$ with $L < t_1 < \cdots < t_K < U$, let $\mathbb{G}_0$ be the space of twice-continuously differentiable functions g on (L, U) such that the restriction of g to each of the intervals $(L, t_1]$, $[t_1, t_2], \ldots, [t_{K-1}, t_K]$,

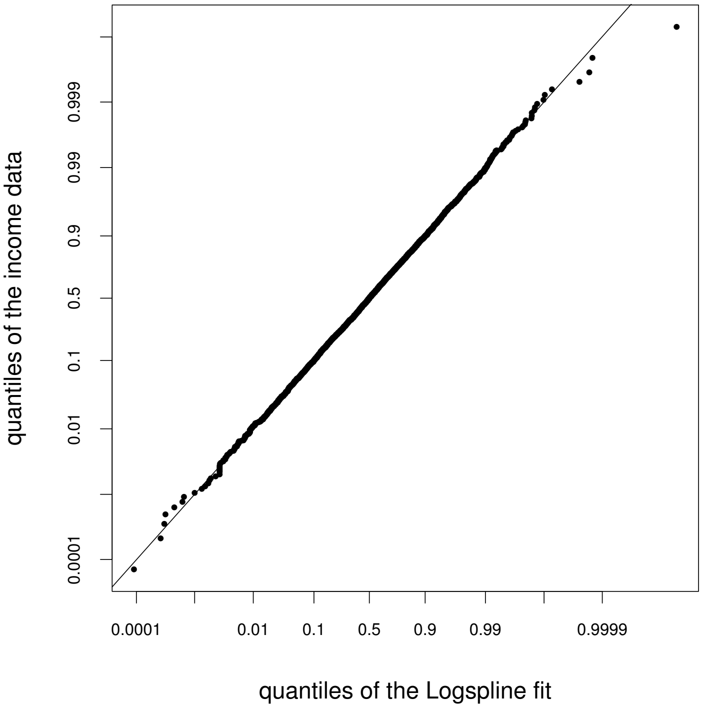

FIGURE 6.6. Probability-probability plot for the income data and the Logspline estimate of the income density.

$[t_K, U)$ is a cubic polynomial. The space $\mathbb{G}_0$ is $(K+4)$-dimensional, and the functions in this space are referred to as cubic splines having (simple) knots at $t_1, \ldots, t_K$. Let $\mathbb{G}$ be the subspace of $\mathbb{G}_0$ consisting of the functions in $\mathbb{G}$ that are linear on $(L, t_1]$ and on $[t_K, U)$. The space $\mathbb{G}$ is K-dimensional, and the functions in this space are referred to as natural (cubic) splines. Set $p = K - 1$. Then $\mathbb{G}$ has a basis of the form $1, B_1, \ldots, B_p$. Right now we assume that the knots $t_1, \ldots, t_K$ are given. In Section 6.2.4 we will examine stepwise algorithms for selecting the knots.

A column vector $\boldsymbol{\theta} = [\theta_1, \ldots, \theta_p]^{\mathrm{T}} \in \mathbb{R}^p$ is said to be *feasible* if

$$\int_L^U \exp\big(\theta_1 B_1(y) + \cdots + \theta_p B_p(y)\big)\, dy < \infty \tag{6.2.1}$$

or, equivalently, if (i) either $L > -\infty$ or $\lim_{y\to-\infty} \sum \theta_j B_j(y) = -\infty$ and (ii) either $U < \infty$ or $\lim_{y\to\infty} \sum \theta_j B_j(y) = -\infty$. We will discuss the choice of basis functions and feasibility of coefficient vectors further in Section 6.2.2.

Let Θ denote the collection of feasible column vectors. Given $\boldsymbol{\theta} \in \Theta$, set

$$C(\boldsymbol{\theta}) = \log\Big(\int_L^U \exp\big(\theta_1 B_1(y) + \cdots + \theta_p B_p(y)\big)\, dy\Big) \tag{6.2.2}$$

and

$$f(y;\boldsymbol{\theta}) = \exp\big(\theta_1 B_1(y) + \cdots + \theta_p B_p(y) - C(\boldsymbol{\theta})\big), \qquad L < y < U. \quad (6.2.3)$$

(We refer to $C(\cdot)$ as the *normalizing constant.*) Then $f(\cdot;\boldsymbol{\theta})$ is a positive density function on (L,U) for $\boldsymbol{\theta} \in \Theta$. The corresponding distribution function $F(\cdot;\boldsymbol{\theta})$ and quantile function $Q(\cdot;\boldsymbol{\theta})$ are given by

$$F(y;\boldsymbol{\theta}) = \int_L^y f(z;\boldsymbol{\theta})\,dz, \qquad L < y < U,$$

and

$$Q(p;\boldsymbol{\theta}) = F^{-1}(p;\boldsymbol{\theta}), \qquad 0 < p < 1$$

(so that $F(Q(p;\boldsymbol{\theta});\boldsymbol{\theta}) = p$ for $0 < p < 1$ and $Q(F(y;\boldsymbol{\theta});\boldsymbol{\theta}) = y$ for $L < y < U$). If $U = \infty$, then the density function is exponential on $[t_K, \infty)$; if $L = -\infty$, then the density function is exponential on $(-\infty, t_1]$.

Let Y be a random variable having a continuous and positive density function. Let $Y_1, \ldots Y_n$ be independent random variables having the same distribution as Y. The log-likelihood function corresponding to the Logspline family, given by

$$l(\boldsymbol{\theta}) = \sum \log f(Y_i;\boldsymbol{\theta}) = \sum_i \sum_j \theta_j B_j(Y_i) - nC(\boldsymbol{\theta}), \qquad \boldsymbol{\theta} \in \Theta, \quad (6.2.4)$$

is strictly concave on Θ. The maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ is obtained by maximizing the log-likelihood function. Since the log-likelihood function is strictly concave, the maximum likelihood estimate is unique if it exists.

Let D be the $n \times K$ matrix having entry $B_{j-1}(Y_i)$ in row i and column j for $1 \le i \le n$ and $2 \le j \le K$ and entry 1 in every row of column 1. If the matrix D has rank K, then the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ exists. We refer to $\hat{f} = f(\cdot\,;\widehat{\boldsymbol{\theta}})$ as the *Logspline density estimate.*

Let $\mathbf{H}(\boldsymbol{\theta}), \boldsymbol{\theta} \in \Theta$, denote the Hessian of $C(\boldsymbol{\theta})$, the $p \times p$ matrix the entry in row j and column k of which is given by

$$\frac{\partial^2 C(\boldsymbol{\theta})}{\partial\theta_j \partial\theta_k} = \int_L^U B_j(y)B_k(y)f(y;\boldsymbol{\theta})\,dy - \int_L^U B_k(y)f(y;\boldsymbol{\theta})\,dy \int_L^U B_j(y)f(y;\boldsymbol{\theta})\,dy, \quad (6.2.5)$$

which matrix is positive definite. Consequently, the function $C(\cdot)$ is strictly convex. Let $Y_1, \ldots Y_n$ be a random sample of size n from f and let $S(\boldsymbol{\theta})$ be the score function; that is, the $p-$dimensional vector of entries

$$\frac{\partial l}{\partial\theta_j}(\boldsymbol{\theta}) = b_j - n\frac{\partial C}{\partial\theta_j}(\boldsymbol{\theta}), \quad (6.2.6)$$

where the sufficient statistics $b_1, \ldots b_j$ are defined by

$$b_j = \sum B_j(Y_i),$$

while

$$\frac{\partial C(\boldsymbol{\theta})}{\partial \theta_j} = \int_L^U B_j(y) f(y; \boldsymbol{\theta})\, dy. \tag{6.2.7}$$

Note that $\partial C(\boldsymbol{\theta})/\partial \theta_j$ is the expected value of $B_j(y)$ relative to the measure having density $f(\cdot; \boldsymbol{\theta})$, so that $\partial l/\partial \theta_j$ is the difference between n times the empirical mean of $B_j(y)$ and the fitted mean relative to the indicated measure. Similarly, the Hessian of $C(\boldsymbol{\theta})$ can be interpreted as the covariance matrix of the basis functions relative to this measure.

6.2.2 Basis functions

The Logspline space that is spanned by $B_1, \ldots, B_{K-1}$ differs in a few aspects from the general one-dimensional B-spline basis that is described in Chapter 2:

- the constant function 1 is not in this space;
- some of the basis functions are nonzero on the intervals $[L, t_1]$ and $[t_K, U]$, even when L is $-\infty$ or U is ∞.

Among bases with these property that span a spline space as described in Section 6.2.1, the Logspline basis has the fewest number of nonzero basis functions for any y. The Logspline basis is defined as follows:

- The first basis function B_1 is linear and strictly decreasing on $[L, t_1]$ and is 0 on $[t_3, U]$. It is not hard to establish that, except for a positive multiplicative factor, this uniquely defines B_1. Curiously,

  $$B_1(y) = a_1 \int_y^{t_3} \int_u^{t_3} B^l(v; t_1, t_2, t_3)\, dv\, du$$

  for some $a_1 > 0$, where $B^l(y; t_1, t_2, t_3)$ is the *linear* B-spline defined by the three knots t_1, t_2 and t_3.

- The second basis function B_2 is linear and strictly increasing on $[t_K, U]$ and is 0 on $[L, t_{K-2}]$. Except for a positive multiplicative factor, this defines B_2. Explicitly,

  $$B_2(y) = a_2 \int_{t_{K-2}}^{y} \int_{t_{K-2}}^{u} B^l(v; t_{K-2}, t_{K-1}, t_K)\, dv\, du$$

  for some $a_2 > 0$.

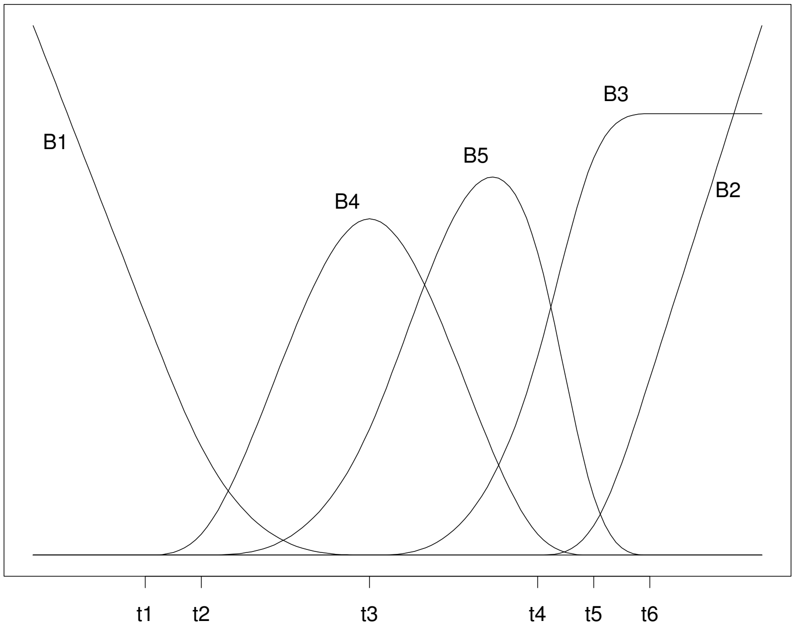

FIGURE 6.7. The Logspline basis.

- If $K > 3$ the third basis function B_3 is 0 on $[L, t_{K-3}]$ and constant on $[t_K, U]$. Except for a positive multiplicative factor this defines B_3. Explicitly,

$$B_3(y) = a_3 \int_{t_{K-3}}^{y} B^q(u; t_{K-3}, t_{K-2}, t_{K-1}, t_K)\, du$$

  for some $a_3 > 0$, where $B^q(x; t_{K-3}, t_{K-2}, t_{K-1}, t_K)$ is the *quadratic* B-spline defined by the four knots $t_{K-3}, \ldots, t_K$).

- If $K > 4$ the jth basis function B_j is the cubic B-spline defined by the five knots $t_{j-3}, \ldots, t_{j+1}$, for $j = 4, \ldots, K-1$.

Figure 6.7 shows the Logspline basis for a situation with $K = 6$. For the Logspline basis, a column vector $\boldsymbol{\theta}$ is feasible if $\theta_1 < 0$ or $L > -\infty$ and $\theta_2 < 0$ or $U < \infty$. For computational reasons it is convenient that feasibility depends only on two of the parameters θ_j. Note that the Logspline basis does *not* have the B-spline property that $\sum_j B_j(y) = 1$ for all y.

We can also define a Logspline basis for $K = 2$ in which case the basis is 1-dimensional. The basis function should be linear on $[L, t_1]$ and $[t_2, U]$, which implies that $B_1(y) = y$. Thus for $K = 2$ feasible Logspline models exist only if $L > -\infty$ or $U < \infty$, and the Logspline densities are exponential densities restricted to $[L, U]$.

6.2.3 Fitting Logspline models

Since the log-likelihood function for Logspline strictly is concave and Logspline models have only a moderate number of parameters, finding the maximum likelihood estimate $\widehat{\boldsymbol{\theta}}$ using a Newton–Raphson algorithm is straightforward (see Chapter 4). There are two complications:

1. to compute the normalizing constant, the Hessian, and the score vector requires numerical integration (see Section 6.7.3); and
2. when $L = -\infty$ or $U = \infty$ there are constraints on θ_1 or θ_2. See Section 6.7.4 for a detailed discussion.

6.2.4 Knot selection

The knot selection methodology involves initial knot placement, stepwise knot addition, stepwise knot deletion, and final model selection based on AIC.

Initially we start with K_{init} knots. Stone, Hansen, Kooperberg, and Truong (1997) use $K_{\text{init}} = \min(2.5n^{\cdot 2}, n/4, N, 25)$, where N is the number of distinct Y_i's (see Section 5.5.1 for a motivation for having a rule of this form). While for many of the methodologies discussed in this book we start with the minimal model (which in case of Logspline would be three knots), we have found that for Logspline a somewhat larger number is needed to give the initial model sufficient flexibility and prevent numerical problems during the first few knot additions. The initial knots are placed according to the rule described in Section 6.7.1. This rule places knots at selected order statistics of the data. The extreme knots are placed at the extreme observations and the interior knots are positioned such that the distances (on an order statistic scale) between knots near the extremes of the data are fairly small and almost independent of the sample size, while the knots in the interior are positioned approximately equidistantly.

The knot addition and knot deletion procedure that is employed for Logspline is essentially the procedure described in Section 3.3.3. In particular, at each addition step of the algorithm we first find a good location for a new knot in each of the intervals (L, t_1), (t_1, t_2), …, (t_{K-1}, t_K), (t_K, U) determined by the existing knots $t_1, \ldots, t_K$. To do this we maximize in each interval the Rao statistic for potential knots located at the quartiles of the data within each interval. The location is then further optimized, which may involve computing a few more Rao statistics [see Section 5.5.2 for an implementation that is used for a number of methodologies and Section 6.7.2 for a couple of details relevant for Logspline]. The search algorithm selects among the best candidates within the various intervals. After a maximum number of knots $K_{\max}$ is reached [Stone, Hansen, Kooperberg, and Truong (1997) uses $K_{\max} = \min(4n^{\cdot 2}, n/4, N)$], we continue with stepwise knot deletion. (For some examples where we knew a priori that there were many modes, we have used $K_{\max} = \min(2.5n^{.5}, n/4, N)$.) During knot deletion we successively remove the least significant knot, where Wald statistics are used to measure significance. We continue this procedure until only three knots are left.

Among all models that are fit during the sequence of knot addition and knot deletion we choose the model that minimizes AIC with default penalty parameter $a = \log n$ (BIC), as described in Chapter 4.

The default values for K_{init} and $K_{\max}$ were developed by Kooperberg and Stone (1991) based on a simulation study. In this study a trade-off was made between how often a Logspline density estimate based on a random sample from a true unimodal density ended up being multimodal and how often a Logspline density estimate based on a random sample from a true bimodal density ended up being unimodal. This rule was then further modified by Kooperberg and Stone (1992) and Stone, Hansen, Kooperberg, and Truong (1997). Kooperberg and Stone (1991) and Kooperberg and Stone (1992) both employed algorithms that involved only stepwise deletion, in which context the size of the largest model is considerably more important than for the algorithm involving stepwise addition and deletion that is described in this chapter. The power $(n^{.2})$ can be motivated to some extent from theory in that the number of knots clearly needs to increase as a small power of n, and this power should equal 1/5 under the assumption that the true density function has bounded second derivative.

6.3 How much to smooth: more examples

We now return to the income data to examine more closely the process of stepwise addition of knots. As it turns out, the density estimate for this data set, even with the initial 15 knots, looks quite reasonable. However, in Figure 6.8 we show part of the sequence of models obtained during the process of stepwise addition when we initially fit a model with 7 knots. The first panel shows the fit with 7 knots and five of the initial knots (the remaining two knots are outside the plotted area), and for the other five panels we show the location of the additional knot. For all but the addition of the 9th knot, which has a Rao statistic of 875, all Rao statistics are between 15 and 60. Note that three of the knots that are added are close to the sharp peak. The third added knot is not close to that peak, but the nonlocal effect of cubic spline knots results in this knot noticeably increasing the height of the peak.

As mentioned earlier, an important issue in density estimation is "how much to smooth?", which, for Logspline, means "how many knots?". When Logspline models are selected using AIC (3.2.29), this translates into "which value of the penalty parameter to use?" As it turns out, for the income data, the fits obtained by Logspline are surprisingly insensitive to the choice of

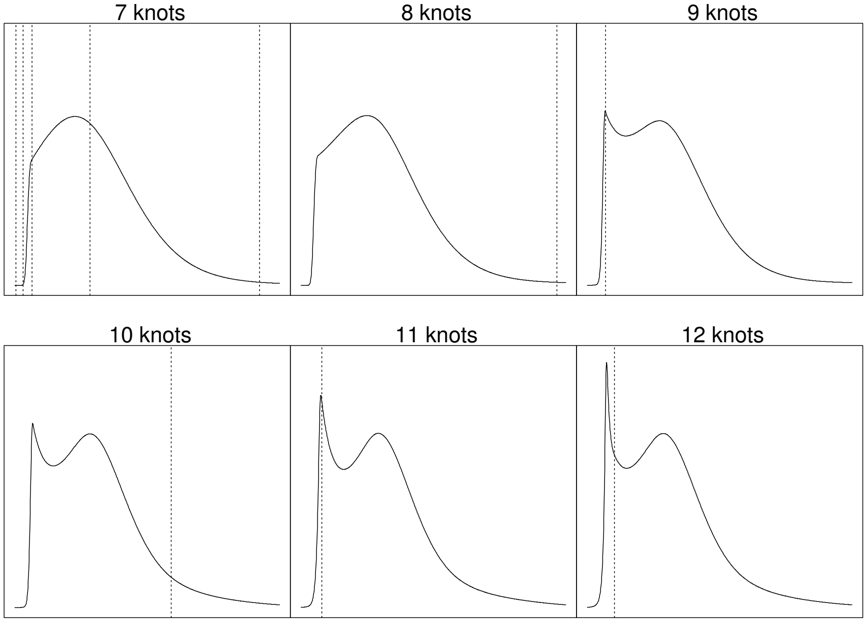

FIGURE 6.8. Logspline density estimate the income data for five steps of knot addition starting with a model with 5 knots. The dashed lines indicate the location of the (new) knots.

the penalty parameter a. In particular, the value of a that was used was the default value $\log 7125 \approx 8.87$. The same fit would have been obtained for any value of a between 5.39 and 9.91; moreover, all values of a between 0.25 and 83.11 yielded very similar Logspline estimates. These estimates have at least three knots close to the sharp peak, with the heights of the peak being very close to that in Figure 6.5. The only difference between these fits is whether some tiny bumps in the tail are present and whether a small bump in the trough between the sharp peak and the second mode is present. This robustness of the methodology is typical for Logspline, especially when applied to larger data sets.

When it is less clear whether modes are present, the penalty parameter may have a larger Influence. A data set that illustrates this issue particularly well is the Mexican stamp data, which has been used extensively in the density estimation literature. What makes this data set interesting is the potentially large number of modes: some references suggest as many as seven modes (Minnotte and Scott 1993; Marron and Chaudhuri 1999). The data set consists of the thicknesses of 485 Mexican stamps printed in 1872–1874 from the 1872 Hidalgo issue. It was first published in Wilson (1983) and brought to the attention of the statistical community in Izenman and Sommer (1988). To allow for Logspline models with many modes, we decided to increase the maximum number of knots $K_{\max}$ from the default

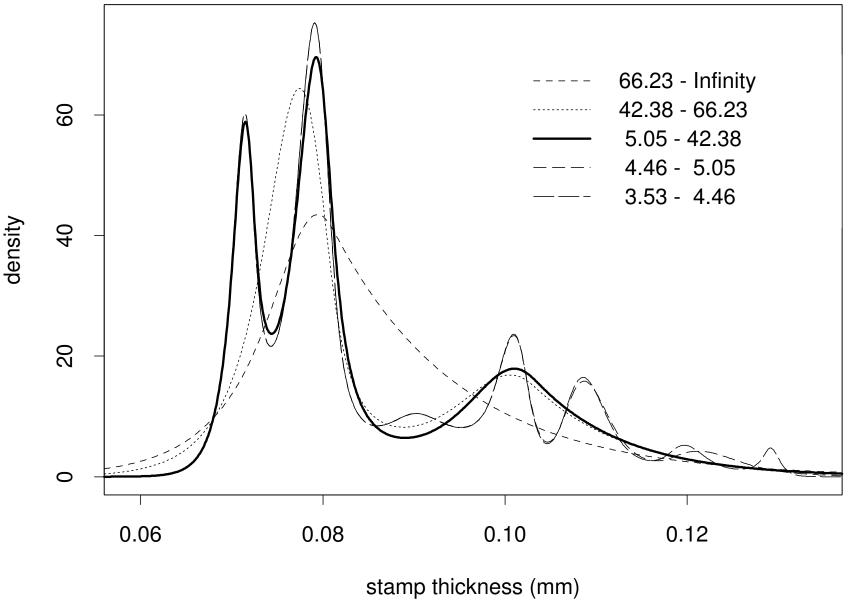

FIGURE 6.9. Logspline density estimates for the stamp data for various values of the penalty parameter a, which we refer to as special models. The special model with solid thick lines uses the default value of a.

value of 15 to 20, and we reduced the minimum distance between knots ($d_{\min}$ in equation (5.5.1)) from the default value of three to two. The thick solid line in Figure 6.9 is the Logspline density estimate with the default penalty parameter $a = \log 485 \approx 6.18$, which has 7 knots and 3 modes. Depending on our choice of a, we could have obtained several other models. In particular, Table 6.1 summarizes the models that were fit during stepwise deletion. As it turned out, with this data set, for every number of knots, the model that was obtained during stepwise deletion had a higher log-likelihood than the model with the same number of knots that was obtained during stepwise addition. Table 6.1 indicates for which values of the penalty parameter a a particular model is optimal.

As we see from Table 6.1, the model with 7 knots would have been selected for any value of a between 5.05 and 42.38, but for smaller values of a models with more knots and for larger values of a models with fewer knots would have been obtained. In Figure 6.9 we also show the density estimates for the models that are indicated by $\triangleleft$ with 3, 5, 13, and 15 knots; these estimates have 1, 2, 6 and 7 modes, respectively. The densities with more than 15 knots have 7 modes and look very similar to the one with 15 knots.

To get a better feeling about the appropriate number of modes, we carried out a small simulation study. We simulated 250 independent samples of size

			minimum	maximum	
number	log-	AIC	penalty	for which	
of knots	likelihood	$a = \log 485$	this model	is selected	
3	1406.57	−2800.78	66.23	Inf	$\lhd$
4	1423.55	−2828.55	NA	NA	
5	1472.81	−2920.87	42.38	66.23	$\lhd$
6	1472.81	−2914.69	NA	NA	
7	1515.18	−2993.26	5.05	42.38	$\prec$ $\lhd$
8	1515.91	−2988.53	NA	NA	
9	1518.71	−2987.95	NA	NA	
10	1519.28	−2982.90	NA	NA	
11	1523.95	−2986.06	NA	NA	
12	1525.43	−2982.84	NA	NA	
13	1530.32	−2986.43	4.46	5.05	$\lhd$
14	1530.36	−2980.32	NA	NA	
15	1534.78	−2982.98	3.53	4.46	$\lhd$
16	1536.55	−2980.33	1.60	3.53	
17	1537.35	−2975.75	0.49	1.60	
18	1537.59	−2970.05	0.44	0.49	
19	1537.81	−2964.30	0.00	0.44	
20	1537.81	−2958.12	0.00	0.00	

TABLE 6.1. Summary of the Logspline models for the stamp data. The symbol $\lhd$ indicates the special models shown in Figure 6.9. The symbol $\prec$ indicates the model selected by BIC.

485 from each of the five densities shown in Figure 6.9, which we refer to as the *special models*. To each of these 250 simulated samples we fit Logspline models with penalty parameters 2, 4, 5, $\log(485) \approx 6.18$, 10, 50, and 100, which we refer to as the *simulated estimates* corresponding to the special model and specified penalty parameter. For each such simulated estimate we then counted the number of modes. Table 6.2 summarizes how often a simulated estimate has fewer modes or more modes than the Logspline fit to the stamp data with the same penalty parameter.

From this table we note that it is extremely unlikely that the "true" density of the stamp data behaves like the the special model in Figure 6.9 with 1 mode (3 knots), the reason being that simulated estimates corresponding to this special model with various penalty parameters do not look like the Logspline fit to the stamp data with that penalty parameter. In particular, although the Logspline fit to the stamp data with the default penalty parameter $a = \log(485)$ has 3 modes, all but two of the simulated estimates with that penalty parameter have fewer than 3 modes; the fitted density with $a = 50$ has 2 modes, but all of the corresponding simulated estimates have fewer than 2 modes; the fitted density with $a = 5$ has 6 modes, but all of the corresponding simulated estimates have fewer than 6 modes; and

penalty parameter a	2	4	5	$\log 485$	10	50	100
modes for stamp data	7	7	6	3	3	2	1
Simulated from the special model having	Simulated estimates have fewer nodes than the fitted density to the stamp data with the same penalty parameter						
1 mode (3 knots)	250	250	250	248	249	250	0
2 modes (5 knots)	249	250	250	215	243	76	0
3 modes (7 knots)	248	249	250	0	0	55	0
6 modes (13 knots)	216	248	194	0	0	48	0
7 modes (15 knots)	181	218	158	7	13	52	0
Simulated from the special model having	Simulated estimates have more nodes than the fitted density to the stamp data with the same penalty parameter						
1 mode (3 knots)	0	0	0	0	0	0	0
2 modes (5 knots)	0	0	0	2	0	0	6
3 modes (7 knots)	0	0	0	26	7	35	6
6 modes (13 knots)	1	0	2	149	42	47	1
7 modes (15 knots)	16	3	20	150	60	57	2

TABLE 6.2. Number of times out of 250 that Logspline estimates with a specified penalty parameter based on data sets simulated from the special models in Figure 6.9 contain fewer (top half) or more (bottom half) modes than the Logspline estimate with the same penalty parameter based on the stamp data.

so on. The same reasoning rules out the true density from behaving like the special model with 2 modes (5 knots). The special model obtained by applying the default penalty parameter $\log 485$ to the stamp data has 3 modes (7 knots). The numbers of modes of corresponding simulated estimates are consistent with those based on the stamp data when $a \geq \log 485$, but inconsistent (too few) for smaller values of a. The special model with 6 modes (13 knots) is not consistent with the corresponding simulated estimates for $a = 4$, but it is consistent or reasonably so for the other values of a. Thus, perhaps the true density of the stamp data behaves like the special model with 7 modes (15 knots). On the other hand, the Logspline fit to the stamp data with penalty parameter $a = 2$ or penalty parameter $a = 4$ has 7 modes. However, in 181 out of 250 simulations for $a = 2$ and 218 out of 250 simulations for $a = 4$, the corresponding simulated estimates have fewer than 7 modes. These numbers do not rule out the true density from having 7 modes, but they are definitely worrisome. In addition, we have observed that changing some of the options of Logspline can yield substantial changes in the fit to the stamp data. Our conclusion is that there is not enough stamp data for Logspline accurately to determine the number of modes of the true density.

In our experience it is fairly unusual for the Logspline estimate to depend so strongly on the penalty parameter and for the default value of a not to seem to give a good estimate. Further examination of the stamp data set reveals that the data are severely rounded (they have only three significant digits), so they certainly could be considered as interval censored. Among the 485 points in this data set, there are only 62 distinct values, and some of the modes and valleys in the fits with 13 and 15 knots describe features that may model observations only in the neighborhood of two consecutive distinct values in the data set. More generally, this example illustrates the desirability of examining Logspline estimates for different values of the penalty parameter.

6.4 Free knot splines and inference

A problem with polynomial spline methods as described in this monograph is that traditional inferential procedures (using the inverse of the Hessian matrix as the covariance matrix) carry out inference conditional on the form of the basis functions and the location of the knots. As such, the adaptivity in selecting the form of these basis functions, which arguably is a strength of polynomial spline methodologies, is ignored. There are roughly two ways around this: use the bootstrap and/or simulation approach (see later in this section, as well as the examples in Sections 7.2.5, 7.3.1, and 7.3.2); use free knot splines.

6.4.1 Free knot splines

In free knot spline methodology, the locations of the knots are treated as additional parameters to be estimated by maximum likelihood along with the other parameters. Logspline density estimation with free knots is discussed in Kooperberg and Stone (2001b). Computing the maximum likelihood estimates is a highly nontrivial problem since the likelihood function (6.2.4) is severely multimodal when the knots $t_1, \ldots, t_K$ are parameters, and degenerate solutions exist when too many of the knots t_j get close together. Kooperberg and Stone (2001b) discuss these numerical issues in detail. Chapter 12 discusses theoretical properties of function estimation with free knot splines. Lindstrom (1999) discusses linear regression using polynomial splines with free knots. In the approaches of both Lindstrom (1999) and Kooperberg and Stone (2001b), the use of the Jupp (1978) transform is critical. This transformation replaces the knots $t_1, \ldots, t_K$ by new parameters

$$h_j = \log \frac{t_{k+1} - t_k}{t_k - t_{k-1}}, \qquad 1 \le j \le K.$$

Here $t_0 = L$, and $t_{K+1} = U$. For Logspline density estimation it is still necessary to select K. Kooperberg and Stone (2001b) use AIC (Chapter 4) with penalty parameter $a = 2$ for this.

With the use of free knot splines, it is straightforward to obtain approximate (pointwise) confidence intervals using the inverse of the Hessian (which is now of dimension $2K+1$) as the covariance matrix. Unfortunately, however, Kooperberg and Stone (2001b) conclude, based on a simulation study, that the coverages of nominal 95% confidence intervals using the free knot algorithm, while closer to 95% than the coverages ignoring knot selection, are still well below 95%.

6.4.2 The bootstrap

Alternatively, we can employ the bootstrap in combination with either the stepwise knot deletion algorithm of Kooperberg and Stone (1992) or the stepwise addition and deletion algorithm of Stone, Hansen, Kooperberg, and Truong (1997) to obtain confidence intervals corresponding to Logspline density estimates. Here we use the former algorithm and examine the coverage of bootstrap percentile intervals (Efron and Tibshirani 1993) for both the log-density and the distribution function. Thus, we take B (we need $B \geq 1000$) samples $\mathbf{Y^i}$ with replacement of size n from the data $Y_1, \ldots, Y_n$, and for each sample $\mathbf{Y^i}$ we obtain the Logspline density estimate. The 95% pointwise confidence interval for $\log \widehat{f}(y)$ is then from the 2.5th to the 97.5th percentile of the B bootstrap estimates for the log-density.

Clearly, the bootstrap is a computationally time consuming procedure for getting confidence intervals, since we need to fit B Logspline densities. However, it is still slightly faster than using the algorithm developed in Kooperberg and Stone (2001b) for fitting Logspline densities with free knots.

A considerably cheaper approach is to hope that the Logspline estimates of the log-density have approximately a normal distribution, but that the estimates of the standard errors that are obtained using standard techniques are too small. If so, we can get by with a much smaller number B of bootstrap estimates (say $B = 25$) by using these estimates to obtain pointwise bootstrap estimates of $\mathrm{SE}^B(\log \widehat{f}(y))$ and transform those back to obtain confidence intervals for f.

6.4.3 A comparison

We now summarize the simulation results from Kooperberg and Stone (2001a,2001b) , in which we generated 250 samples of size 250 and 250 samples of size 1000 from each of four distributions:

Normal 2 A mixture of two normal distributions, so that the true density of Y is given by

$$f(y) = c\Big(\frac{1}{3}f_{Z_1}(y) + \frac{2}{3}f_{Z_2}(y)\Big)\mathrm{ind}(-4, 8),$$

where Z_1 has a normal distribution with mean 0 and standard deviation 0.5, Z_2 has a normal distribution with mean 2 and standard deviation 2, ind$(\cdot)$ is the usual indicator function, and c is the multiplier to correct for the truncation to $(-4, 8)$.

Normal 4 As in example 1, but the mean of Z_2 is 4 and Y is truncated to $(-2, 10)$.

Normal 6 As in example 1, but the mean of Z_2 is 6 and Y is truncated to $(-1.5, 12)$.

Gamma 2 A gamma distribution with shape parameter 2 and mean 1, with Y truncated to the interval $(0, 9)$.

The Normal 2 density has one mode, but a clear second hump; Normal 4 has two, not very well separated, modes; Normal 6 has two well separated modes; and the Gamma 2 density is unimodal.

In Table 6.3 we compare the coverages of four approaches for getting confidence intervals. Two columns use the free spline methodology. These columns are the coverages obtained by using the $(2K + 1) \times (2K + 1)$ Hessian that treats knots as free parameters, labeled "SE", and the one using the $(K+1)\times(K+1)$ Hessian that assumes the knots are fixed, labelled "SEFX". As can be seen from this table, the coverages are well below the nominal 95% level. The last two columns are using bootstrap samples for the logspline density estimation procedure of Kooperberg and Stone (1992). The third column is based on 1000 bootstrap samples, and the confidence intervals are from the 2.5th through the 97.5th (pointwise) percentiles. For the fourth column we generated only 25 bootstrap samples, computed the pointwise standard errors for the log-density, and then transformed those to obtain the confidence intervals.

It is clear from this table that the confidence intervals based on the free knot spline standard error or the fixed knot spline standard error have too low coverage. Surprisingly, the coverages for the bootstrap percentile intervals are consistently too high. It is our impression that this is due to some instability in the stepwise logspline algorithm when there are many repeat observations, causing the intervals to be occasionally too large. This is in line with what we will see for the income data in the next section. Interestingly, the coverages in the last column of Table 6.3 corresponding to the bootstrap SE approach, are not only very close to 95% on average, but also show little variation. Thus, our current approach to inference for Logspline density estimates would be to use this last approach.

		Free Knots		Bootstrap	
	Density	SE	SEFX	Percentiles	SE
$n = 250$	Normal 2	84.0	77.4	97.4	95.2
	Normal 4	88.8	82.5	97.4	96.4
	Normal 6	89.0	84.0	96.5	94.6
	Gamma 2	86.2	81.2	97.8	97.3
$n = 1000$	Normal 2	89.2	79.6	96.8	94.4
	Normal 4	89.3	82.7	98.0	94.7
	Normal 6	86.2	81.4	96.3	92.9
	Gamma 2	84.0	77.3	97.4	95.4
	Average	87.1	80.7	97.2	95.1

TABLE 6.3. Coverages for four different approaches to obtaining confidence intervals for a log-density, estimated using logspline.

6.5 Censoring and truncation

6.5.1 *The Fyn diabetes data*

The Fyn diabetes data, which is extensively discussed in Andersen, Borgan, Gill, and Keiding (1993) consists of data on the 1499 diabetes patients that lived on July 1, 1973, in the County of Fyn in Denmark. The data was collected by Green, Hauge, Holm, and Rasch (1981). For all patients it was known whether they were still alive and living in Fyn on December 31, 1981, and, if not, whether and when they died or moved. We also know the age of each patient, as well as the age at which diabetes for this patient was diagnosed. There are 783 men, of whom 254 died, and 716 women, of whom 237 died. It is of interest to see how the survival distribution for the Fyn diabetics differs from the general Danish population. Therefore we would like to estimate the density of the age at dying of the Fyn diabetics. There are two complications that we need to deal with before we can apply Logspline.

- At the end of the study the majority of the patients were still alive. Clearly, when someone is still alive on December 31, 1981, and is, say, 73 years old, we know something: the age of this person at dying is at least 73 years. This phenomenon is called (right-)censoring.

- Not all patients enter the study at the same age. In particular, potential participants who have died before the study started are not included. Thus, if we consider the year when a participant was born as random, among all diabetics ever alive in Fyn, patients who achieve old age have a larger probability of actually being part of the study than those who die at young age. Ignoring this sampling mechanism induces bias in our estimate. This type of selection of participants is called (left-)truncation.

6.5.2 Implications for Logspline

The regular Logspline approach

Before we continue our analysis of the Fyn diabetes data, we first discuss the extensions of Logspline needed to deal with censored and truncated data. Censoring and truncation are discussed in more detail in the chapter about survival analysis (Chapter 7). A brief discussion is provided in this section, but readers unfamiliar with these concepts may want to read the more detailed discussion in Section 7.1.2. Logspline density estimation with censored data has previously been discussed in Kooperberg and Stone (1992); this is the first published discussion for an extension to Logspline for truncated data.

A random variable Y is said to be censored if the exact value of Y is unknown, but it is known that $Y \in A$ for some interval $A \subset \mathbb{R}$. In survival analysis it is not uncommon that at the end of a study some of the participants are still alive, as is the case for the Fyn diabetics study. If this is the case for a particular participant, we know that the participant has at least survived until the end of the study, which is at, say, time C. Since we know that eventually this participant would have died, we know that $Y \in A = (C, \infty)$. This particular form of censoring is called *right-censoring*, and we usually will say that Y is censored at C. For *interval-censoring* we know that $Y \in A = [A_1, A_2]$ (or, maybe $Y \in A = (A_1, A_2]$). Interval censoring occurs, for example, when participants in a medical study are examined periodically, and at some examination it is known that a participant did not have a certain disease at the previous examination, but that the participant does have the disease at the current examination, thus the participant contracted the disease sometime between these two examinations. Rounding of data is also a form of interval-censoring[1]

Truncation occurs when a random variable Y (which may or may not be censored) is only included in the sample if it is in some interval B. The most common form of truncation is left-truncation. This happened for the Fyn diabetics study, where only those diabetics who were alive when the study started could be included. If a person had age T at the beginning of the study then $B = [T, \infty)$, and we say that Y is (left-)truncated at time T.

[1] The stamp data, discussed in Section 6.3 can be analyzed as an interval-censored data set by Logspline. Since the censoring intervals are quite narrow, the results for Logspline are very similar to those described earlier.

If a random variable Y is censored to be in A and truncated to be in B the likelihood of Y is $\int_{A\cap B} f_Y(y)dy / \int_B f_Y(y)dy$ if $A \cap B$ has positive length, $f_Y(A)/\int_B f_Y(y)dy$ if A consists of a single point, which is in B, and 0 if $A \cap B = \emptyset$. In particular, the contribution to the log-likelihood of a random variable Y that may be censored and is truncated at time T is identical to the contribution of an untruncated random variable Y minus the contribution of an independent untruncated random variable that is right-censored at time T.

In principle it appears straightforward to design a Logspline procedure that can deal with censored and truncated data. In practice there are a number of complications.

- When there is censoring or truncation, the log-likelihood function for Logspline is not guaranteed to be concave. In our experience this is usually not a problem. We suspect that the log-likelihood function is often close to being concave. In addition, when we add a basis function, we already start from the maximum of the likelihood function in a one-dimensional smaller subspace, usually a pretty good starting value. Nevertheless, to deal with possible nonconcavity, we recommend checking regularly during the Newton–Raphson iterations whether $\mathbf{H}$ is negative definite, and, if at some stage $\mathbf{H}$ fails to be negative definite, replacing it by $\mathbf{H}' = \mathbf{H} - a\mathbf{I}$, where a is slightly larger than the largest eigenvalue of $\mathbf{H}$ if this eigenvalue is positive (see Kennedy and Gentle 1980, Section 10.2.2, and Kooperberg and Clarkson 1997), or to use an off-the-shelf optimizer that does not require the Hessian to be concave at every iteration, such as the S-Plus function `nlminb()`, which is based on the trust-region approach developed by Gay (1983).

- When a large amount of data is censored, it is not clear whether the (effective) sample size, as used for such parameters as the maximum number of knots $K_{\max}$ and the AIC penalty parameter a, is really still n. For example, a right-censored observation at L contains no information about f. We defer discussion of this issue to Chapter 7, but remark that in some survival analysis applications (e.g. power calculations) the driving force is the number of uncensored observations.

- With censored and truncated data the expressions for the Hessian and the score function become considerably more complicated and many more numerical integrals need to be computed. See Kooperberg and Stone (1992).

Using the Kaplan–Meier estimate

Another approach to Logspline density estimation with possibly right-censored and left-truncated data was discussed in Koo, Kooperberg, and

Park (1999). The main idea behind this approach is that for complete (uncensored and untruncated) data the score equations (6.2.6) that are solved to obtain the MLE $\widehat{\boldsymbol{\theta}}$ can be written as

$$\int_L^U B_k(y) f(x; \widehat{\boldsymbol{\theta}}) dy = \int_L^U B_k(y) d\widehat{F}(y), \tag{6.5.1}$$

where $\widehat{F}(y)$ is the usual empirical distribution function. When some of the data is right-censored or left-truncated, we can use the product-limit (Kaplan–Meier) estimator $\widetilde{F}(y)$ to obtain an estimate of the distribution function[2](Andersen, Borgan, Gill, and Keiding 1993).

Assume that we observe (X_i, δ_i, T_i), $i = 1, \ldots, n$, such that if $\delta_i = 1$ then $Y_i = X_i$ is an independent sample of $F(y|y > T_i)$, while if $\delta_i = 0$ Y_i is censored at X_i and truncated at T_i. The product limit estimator then takes the form

$$\widetilde{F}(y) = \sum_{i:\delta_i = 1} v_i I(y \geq Y_i),$$

with $v_i > 0$ with $\sum v_i \leq 1$. Note that if all Y are uncensored and untruncated then $v_i = 1/n$. The idea is to replace (6.5.1) by

$$\int_L^U B_k(y) f(x; \widehat{\boldsymbol{\theta}}) dy = \int_L^U B_k(y) d\widetilde{F}(y), \tag{6.5.2}$$

which means that we can fit a Logspline model using the algorithm for uncensored data, by using case weights v_i for the Y_i.

Since for data sets that contain many truncated or censored observations the v_i may be very different (although typically v_i and v_j are similar when Y_i and Y_j are close) the Rao and Wald statistics as well as the model selection using AIC need to be modified. For example, an increase in the log-likelihood of 1 for the addition of a knot in a region where the typical v_i is 0.1 is much more impressive than an identical increase in the log-likelihood for the addition of a knot in a region where the typical v_i is 10. See Koo, Kooperberg, and Park (1999) for details.

Comparison

Clearly, both approaches to Logspline with censored data have their advantages. The regular Logspline approach is applicable with interval-censoring or forms of truncation other than left-truncation. This approach also does not require any "fudging" with the model selection, as the approach using the Kaplan–Meier estimate does. On the other hand, the approach using the Kaplan–Meier estimate is much faster, since it does not require that all

[2] If all observations are left-truncated, this is formally only an estimate of the conditional distribution function $F(y|y > T_{\min})$, where $T_{\min}$ is the minimum of the truncation times; we ignore this distinction.

the additional integrals involved with censoring or truncation be computed. Users with a more traditional survival analysis background might find the relation with the Kaplan–Meier estimate appealing. Koo, Kooperberg, and Park (1999) establish some theoretical rates of convergence, which are not available for regular Logspline with censored data.

6.5.3 The Fyn diabetes data analyzed

For each person the data set contains, among other information, the age at which the person enrolled in the study (T_i) and the age at which the person left the study (Y_i), either because the participant moved (censoring), the study ended (censoring) or the participant died (uncensored). The participants who stayed in the study until the end were followed for seven and a half years. For each participant we know the gender and whether the participant was censored ($\delta_i = 0$) or died ($\delta_i = 1$). The survival data Y_i is left-truncated by the age at which the participant entered the study T_i.

It is of interest to assess how different the survival functions for the Fyn diabetics are from the general Danish population. On the left side of Figure 6.10 we show the Logspline density estimate of the survival function of Fyn diabetics based upon the left-truncated sample (Y_i, δ_i, T_i) using the regular Logspline methodology, as well as the estimate using the Logspline methodology with the score equations based on the Kaplan–Meier estimate. As a comparison, we have drawn the survival function based upon the Danish vital statistics for 1975 and the Kaplan–Meier estimate for the survival distribution among the Fyn diabetics. The youngest death in the Fyn data is a 19 years old woman. Formally we can thus only estimate the survival distribution of the Fyn diabetics, conditional on surviving until age 19. However, since few people die before that age, we ignore this distinction. Still, we note that at about age 20 years the survival function based on the Danish vital statistics is indeed slightly lower than those corresponding to the Fyn diabetics - implying that a small percentage of the population indeed does die before age 19. We also notice that the three survival functions based on the Fyn data appear almost identical, suggesting that both Logspline procedures provide good fits to the data. The Logspline estimate using the Kaplan–Meier estimate follows, not surprisingly, the Kaplan–Meier curve more closely. Moreover, the survival function based on the Danish vital statistics is clearly very different from those based on the Fyn diabetics data.

On the right side of Figure 6.10 we show density estimates for the Fyn diabetes data using both approaches for Logspline to dealing with censored and truncated data. Interestingly, we note that each estimate yields a bimodal distribution, suggesting that certain diabetics (e.g. some of those having juvenile diabetes) tend to die at a younger age than the other diabetics. We note that the two Logspline estimates are fairly similar.

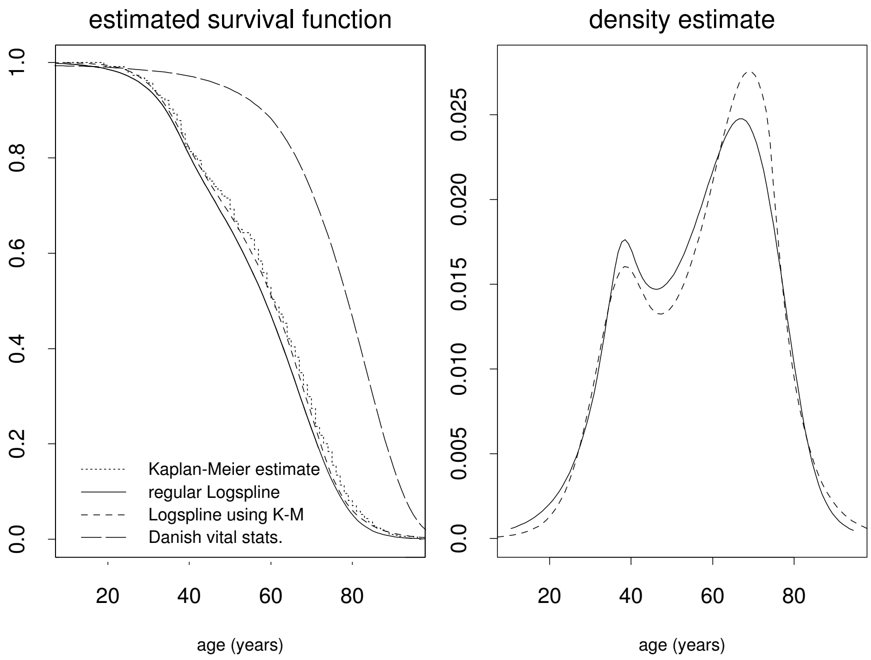

FIGURE 6.10. Logspline density estimate and survival function estimate for the Fyn diabetics data.

To show that ignoring either truncation or censoring leads to erroneous estimates, we applied the regular Logspline procedure to the Fyn data ignoring either the censoring (by assuming that patients die at the censoring time), the truncation, or both. The estimates of the survival function and the density function are given in Figure 6.11. As we can see, ignoring the proper sampling mechanism gives erroneous results, where accidentally the effect of the censoring and truncation roughly offset each other for this data set.

6.6 Multivariate density estimation

Multivariate Logspline density estimation brings with it a number of complications. A first approach to multivariate density estimation, compatible with other extended linear models, would be to use either linear or cubic splines, and to use an ANOVA decomposition type algorithm to build a Logspline model. For bivariate density estimation this would lead to the following model:

$$f(y_1, y_2; \boldsymbol{\theta}) = \exp\big(\theta_1 B_1(y_1, y_2) + \cdots + \theta_p B_p(y_1, y_2) - C(\boldsymbol{\theta})\big),$$

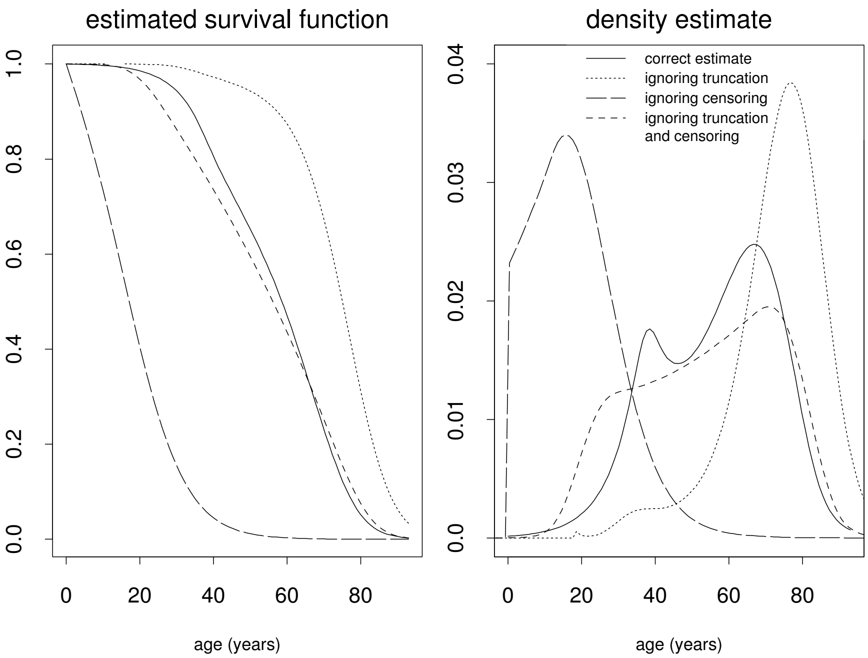

FIGURE 6.11. Logspline density estimate and survival function estimate for the Fyn diabetics data ignoring the correct sampling scheme.

with the normalizing constant

$$C(\boldsymbol{\theta}) = \log\Big(\int_{L_1}^{U_1}\int_{L_2}^{U_2} \exp\big(\theta_1 B_1(y_1,y_2) + \cdots + \theta_p B_p(y_1,y_2)\big)\, dy_1\, dy_2\Big).$$

Even for linear splines this integral can no longer be computed analytically, but it can be reduced to a univariate integral that is closely related to the exponential integral (Abramowitz and Stegun 1965). For cubic splines this is no longer possible, and genuine multidimensional numerical integrals need to be computed.

The requirement that $\boldsymbol{\theta}$ be feasible (6.2.1) may now result in constraints on all basis functions that do not vanish on a bounded interval. For bivariate density estimation this may lead to as many as $2k_1+2k_2$ constraints and for trivariate density estimation as many as $O(2k_1k_2+2k_1k_3+2k_2k_3)$ constraints, where k_i is the number of knots in the ith dimension. In practice this means that for anything but bivariate Logspline density estimation, we need to fix integration bounds.

Probably for these reasons, the methodological development of multivariate Logspline density estimation has been limited. Koo (1996) discusses bivariate Logspline density estimation using cubic splines with finite integration bounds. Kooperberg (1998) uses linear splines to estimate the bivariate density when some of the data may be censored. As neither of these two approaches has publically available software, and implementation of these methods is nontrivial, we omit further discussion. In Chapter 9 we discuss an entirely different approach to bivariate Logspline density estimation, using a triangular spline basis.

6.7 Technical details

6.7.1 Initial knot placement

Kooperberg and Stone (1991) discuss a number of requirements for automatic knot placement for Logspline models. Some of these requirements come from the fact that we want $\hat{f}$ to behave appropriately when the data is transformed linearly. In particular:

- knots should be placed at or near selected order statistics;
- the corresponding indices should be symmetrically distributed about $(n+1)/2$;
- there should be knots at the first and last order statistics;
- the pattern of extreme knots should be approximately independent of sample size, so that the tails are estimated with the same accuracy, independently of sample size;
- the middle knots should be approximately at equally spaced indices, since when we are estimating the middle of the density, it should make no difference whether we are at, say, the 20th percentile or the 60th percentile of the data.

We provide details of the initial knot placement for the situation in which all Y_i are distinct and uncensored. Let K denote the number of knots that we wish to position. The knot placement will be determined by a sequence of numbers $r_1, \ldots, r_K$ such that $1 \leq r_1 < \cdots < r_K \leq n$. Let $1 \leq k \leq K$ and let m denote the greatest integer in r_k, so that $m \leq r_k < m+1$. Then the k^{th} knot will be placed at

$$(m+1-r_k)Y_{(m)} + (r_k - m)Y_{(m+1)} \, .$$

(In particular if $r_k = m$, the k^{th} knot is placed at $Y_{(m)}$.) Our symmetry condition is that

$$r_k + r_{K+1-k} = n+1 \; , \quad 1 \leq k \leq K,$$

which implies that

$$r_{k+1} - r_k = r_{K+1-k} - r_{K-k} \; , \quad 1 \leq k \leq K.$$

In order that there be knots at the first and the last order statistics, we choose $r_1 = 1$ and $r_K = n$.

Set $g_k = r_{k+1} - r_k$ for $1 \leq k \leq K-1$. In order to satisfy the remaining features, we ended up by requiring that

$$g_k = 4 \cdot [(4-\epsilon) \vee 1] \cdot \ldots \cdot [(4-(k-1)\epsilon) \vee 1] \ , \ \ 1 \leq k \leq \frac{K}{2},$$

where $\epsilon \in \mathbb{R}$; here $a \vee b = \max(a,b)$. The constant ϵ is determined as follows: if K is an odd integer, then $2r_{(K+1)/2} = n+1$; if K is an even integer, then $r_{K/2} + r_{K/2+1} = n+1$.

We will now give some examples of our knot placement rule, in which r_k has been rounded off to the nearest integer.

Example 1

$n = 150$, $K = 7$ and $\epsilon \doteq .1881$. The rounded-off values of r_k are as follows:

k	1	2	3	4	5	6	7
r_k	1	5	20	75	131	146	150

Example 2

$n = 150$, $K = 12$ and $\epsilon \doteq 1.2329$. The rounded-off values of r_k are as follows:

k	1	2	3	4	5	6	7	8	9	10	11	12
r_k	1	5	16	33	50	67	84	101	118	135	146	150

Example 3

$n = 500$, $K = 10$ and $\epsilon \doteq .5300$. The rounded-off values of r_k are as follows:

k	1	2	3	4	5	6	7	8	9	10
r_k	1	5	19	60	158	343	441	482	496	500

When several of the Y_i are identical, this rule needs to be modified slightly to prevent knots from coinciding.

6.7.2 Stepwise addition for Logspline

The algorithm for finding the location of a potential new knot for Logspline is described in Section 5.5.2. Note that for Logspline when we compute the Rao statistic for a new knot, we do not recompute the complete basis; instead, we consider a new basis function that is similar to $B_2(y)$ (see Section 6.2.2), and which depends on the new knot, t_{K-1}, and t_K. This way, for each candidate for a new knot only one column of the Hessian and one element of the score function need to be computed, all other elements having already been computed during the most recent set of Newton–Raphson iterations.

6.7.3 Numerical integration

All of the integrals to compute the normalizing constant, Hessian and score function (6.2.2, 6.2.5, 6.2.7) are easily seen to be sums of integrals of the form

$$\int_a^b p_1(x) \exp p_2(x)\, dx,$$

where a is L or one of the knots $t_1, \ldots, t_K$, $b = t_1$ if $a = L$, $b = t_{j+1}$ if $a = t_j$, for $j = 1, \ldots, K-1$, $b = U$ if $a = t_K$, $p_1(x)$ is a polynomial of order 1 for the normalizing constant and at most of order 4 or 7 for every element of the score vector or the Hessian, and $p_2(x)$ is a polynomial of order 1 on the intervals $[L, t_1]$ and $[t_K, U]$ (thus the integrals can be computed analytically on these intervals) and a polynomial of order 4 on each of the intervals $[t_j, t_{j+1}]$. In addition, for all integrals over each of the intervals $[t_j, t_{j+1}]$ the polynomial $p_2(x)$ is the same for the normalizing constant and all elements of the score function and Hessian for fixed $\boldsymbol{\theta}$. Thus, to compute the normalizing constant and all entries of the score vector and the Hessian, it suffices to compute the integrals

$$c_{ij} = \int_{t_j}^{t_{j+1}} x^i \exp\big(\theta_1 B_1(x) + \cdots \theta_p B_p(x)\big)\, dx$$

for $0 \leq i \leq 6$ and $1 \leq j \leq K-1$. The integrals c_{ij} are efficiently computed for all i simultaneously using Gaussian quadrature (Abramowitz and Stegun 1965). We use a limited number of integration points on each interval during the Newton–Raphson iterations, followed by one or two iterations with higher precision when convergence is almost achieved.

6.7.4 Constrained optimization

Depending on L and U, there may be one or two constraints on parameters during the optimization of the form $\theta_i < 0$. These constraints differ from the usual constraints available in commercial optimization routines, which tend to be of the form $\theta_i \leq 0$ (e.g. `nlminb()` in S-PLUS). Based on our experience we currently use a combination of the following two approaches.

1. Initially we reparameterize the coefficients on which there are constraints as $\theta'_j = \log(-\theta_j)$, $j = 1, 2$. There are now no constraints on θ'_j, and new expressions for the score vector and the Hessian are immediately obtained using the chain rule for differentiation. We find this approach easy to use, in particular, since it can be used in combination with off-the-shelf optimization routines. However, when θ_1 or θ_2 is very close to zero, and θ'_1 or θ'_2 are thus very small, numerical problems sometimes prevent convergence. If this happens, we use the second approach, which we have found to be more robust.

2. For the second approach, we replace infinite values for L and U temporarily by finite ones. We describe this algorithm for the situation with two constraints, $L = -\infty$ and $U = \infty$, the situation with one constraint is dealt with similarly:

   (a) Set $L^{(1)} = 2t_1 - t_2$ and $U^{(1)} = 2t_K - t_{K-1}$.

   (b) Replace L by $L^{(1)}$ and U by $U^{(1)}$ in (6.2.2, 6.2.5, 6.2.7). Compute $\widehat{\boldsymbol{\theta}}$ using the (unconstrained) Newton–Raphson algorithm.

   (c) If $\hat{\theta}_1 < 0$ and $\hat{\theta}_2 < 0$ after the previous step,

      i then carry out step (d), using $\widehat{\boldsymbol{\theta}}$ of step (b) as starting value;

      ii else, set $L^{(2)} = 2L^{(1)} - t_1$ and $U^{(2)} = 2U^{(1)} - t_K$. Return to step (b).

   (d) Compute $\widehat{\boldsymbol{\theta}}$, integrating over the whole real line. If, at some intermediate stage $\hat{\theta}_1^{(m)} \geq 0$ or $\hat{\theta}_2^{(m)} \geq 0$, go back to step (c-i), thereby using for L and U the values last used during step (b). If the algorithm converges without this happening, we have obtained $\widehat{\boldsymbol{\theta}}$.

   Note: a cheap alternative to correctly dealing with a constraint $\theta_i < 0$ is to replace this constraint by $\theta_i \leq -\epsilon$ for some small $\epsilon > 0$.

6.8 Notes

Literature

Silverman (1986) is an excellent introduction to density estimation in general and kernel density estimation in particular. Other books on density estimation include Tapia and Thompson (1978).

One of the earliest uses of splines for density estimation is the histospline (Boneva, Kendall, and Stefanov 1971; Wahba 1976). Histosplines are not a true nonparametric density estimation procedure. Rather the histospline is a spline that is fit to a histogram, which then appears to be a smooth density estimate. Boneva, Kendall, and Stefanov (1971) and Wahba (1976) use a penalized likelihood (smoothing spline) approach to histosplines. Later papers (Morandi and Costantini 1989; Minnotte 1998) enforce that the area under the histospline matches the corresponding area under the histogram.

At approximately the same time as the introduction of the histospline, Good and Gaskins (1971) proposed the use of penalized likelihood methods (smoothing splines) for density estimation. Good and Gaskins (1971) uses a penalty of the form $\int \left[(\sqrt{f}\,)''\right]^2$. O'Sullivan (1988a) uses a penalty on the second derivative of the log-density. This would lead to a smoothing spline with a knot at each data point. To circumvent the computational problems he uses a smaller number of knots and B-spline basis functions. A cross-validation argument yields an automatic choice of the smoothing parameter. Gu (1993) uses a similar form of the penalty as O'Sullivan (1988a). He does, however, put a knot at every data point. In our experience that makes his code impractical to use (see the rejoinder of Stone, Hansen, Kooperberg, and Truong 1997). Eilers and Marx (1996) reinvents the procedure of O'Sullivan (1988a). However, where O'Sullivan (1988a) allows knots to be located anywhere, the procedure of Eilers and Marx (1996) requires knots to be equidistant, which makes the algebra (but not the computations!) easier.

Abrahamowicz, Ciampi, and Ramsay (1992) uses polynomial splines for density estimation in an approach that is somewhat similar to Logspline. The main differences are that Abrahamowicz, Ciampi, and Ramsay (1992) fit a polynomial spline to the density, rather than the log of the density, as is done for Logspline. They select the number of knots using AIC. The knots are always equidistant on an order statistics scale. The procedure of Abrahamowicz, Ciampi, and Ramsay (1992) can also deal with right-censored data.

The locfit methodology of Loader (1996), which uses local polynomials with adaptively selected bandwidths, displays similar spatial adaptation as Logspline. There is an enormous literature on kernel density estimation, any list of references would only be incomplete.

History

Stone and Koo (1986) were the first to propose Logspline density estimation. In this paper Logspline models with a few fixed knots, positioned at order statistics of the data, were used. Kooperberg and Stone (1991) used a stepwise deletion algorithm to select knots. They use a preliminary transformation to transform data from $[0, \infty)$ to $\mathbb{R}$. Kooperberg and Stone (1992) abandon this transformation. This paper extends the Logspline methodology to censored data. Stone, Hansen, Kooperberg, and Truong (1997, Sec 5) discusses a Logspline procedure that involves stepwise addition and deletion of knots.

Stone (1989) and Stone (1990) discuss theoretical properties of Logspline models; see also Chapter (11).

Software

There is code available for several versions of the Logspline method. The version that is described in this chapter has been written in C and an interface based on R and S-Plus has been developed. This programs are currently available as part of the `polspline` package from `CRAN`

`http://cran.r-project.org/src/contrib/PACKAGES.html` and from Kooperberg's website

`http://bear.fhcrc.org/∼clk/soft.html`

This version does not deal with censored or truncated data. Insightful Corp. has developed a version of Logspline that does include these capabilities. A link to the appropriate Insightful website is on Kooperberg's website. The `polspline` package also contains a C-program and an interface to S-Plus for the version of Logspline of Kooperberg and Stone (1992). This version uses stepwise deletion of knots from an initial set of starting knots (but no stepwise addition of knots), and it can deal woth censored data but not with truncated data.