COBS: Qualitatively Constrained Smoothing via Linear Programming[1]

Xuming He
Department of Statistics, University of Illinois, Champaign, IL 61820, and Department of Mathematics, National University of Singapore, Singapore 119260

Pin Ng
Department of Economics, University of Illinois, Champaign, IL 61820

Summary

Popular smoothing techniques generally have a difficult time accommodating qualitative constraints like monotonicity, convexity or boundary conditions on the fitted function. In this paper, we attempt to bring the problem of constrained spline smoothing to the foreground and describe the details of a constrained B-spline smoothing (COBS) algorithm that is being made available to S-plus users. Recent work of He & Shi (1998) considered a special case and showed that the L_1 projection of a smooth function into the space of B-splines provides a monotone smoother that is flexible, efficient and achieves the optimal rate of convergence. Several options and generalizations are included in COBS: it can handle small or large data sets either with user interaction or full automation. Three examples are provided to show how COBS works in a variety of real-world applications.

Keywords: Constraint; Information criterion; Knot selection; Linear program; Nonparametric regression; Regression quantile; Smoothing Spline.

1 Introduction

A huge amount of research has been carried out in the past few decades on nonparametric function estimation based on the idea of smoothing. A number of highly successful smoothing methods are available in S-plus. Among them are smoothing splines (Wahba 1990), kernel smoothing (Watson 1966), local-span supersmoother (Friedman 1984), and robust smoothing via Lowess (Cleveland 1979). Several other smoothers are also available from *statlib*. Important recent references to nonparametric smoothing include Härdle (1990), Hastie & Tibshirani (1990), and Green & Silverman (1994).

Data smoothing is often viewed as a graphical method to uncover the underlying relationship between two variables. In some applications, the functions being estimated are known to satisfy certain qualitative properties such as monotonicity. For example in variable transformations, it is often desirable to restrict oneself to monotone functions. Further applications can be found in growth charts, brain image registrations, and probability curve estimation. In other cases, concavity or convexity constraints may be desirable. Examples include cost functions and efficient production frontiers in economics, where the estimated functions are expected to be convex and concave respectively. If the response variable is a proportion, one naturally wants the fitted curves to fall between 0 and 1. For cyclical time series, one might want the fitted curve at the last period to match that at the first period of a cycle. In some cases, the function values or its derivatives at some specific points are known and need to be satisfied by the fitted curve.

Any smoother that performs local averaging over the response values will yield a fitted function falling within the range of the response values. In some applications, this is highly desirable. For example, if the response variable is age or income, zero is the intrinsic lower bound of the estimated function. Conventional spline methods, however, may not preserve this positivity near the boundaries. The constrained B-spline smoothing method we introduce in this paper can easily impose such boundary conditions and help overcome this weakness of spline smoothers.

We will argue that a constrained smoother that incorporates prior information often improves efficiency of the estimators. Delecroix, Simioni & Thomas-Agnan (1995) report a simulation study on this. In the case of monotone smoothing, several methods have been proposed in the literature; see Härdle (1990, Chapter 8) , Hawkins (1994) and Ramsay (1988) for further details. Wright & Wegman (1980) contain a general treatment using splines that includes monotonicity and convexity constraints in least squares regression. Nevertheless, few constrained smoothing algorithms are publi-

[1] The authors would like to thank two anonymous referees, David Scott and Roger Koenker for their insightful comments. Xuming He's research is supported in part by National Security Agency Grant MDA904-96-1-0011 and National Science Foundation Grant SBR 9617278. All the computation in this paper was performed on the Sparc 20 in the Econometric Lab at the University of Illinois supported by NSF Grant SBR-95-12440.

cally available as we write due to the difficulty of incorporating restrictions like those mentioned above.

COBS (COnstrained B-Splines) is a very attractive constrained smoothing method with some unique advantages. Extending the earlier work of Ramsay (1988) and Koenker, Ng & Portnoy (1994), He & Shi (1998) considered a special case of monotone smoothing and laid down the foundation for COBS. The present paper focuses on the algorithmic aspect of COBS and contains a wide variety of options for flexible application.

We begin with a general framework of L_1 minimization for function estimation. This includes two general classes of spline smoothers: smoothing splines (with a roughness penalty) and regression splines (without roughness penalty). Two options are provided in COBS to determine the smoothing parameter of the smoothing splines as well as the knot formation of the regression splines. The first option allows user interaction, which allows users formulate their own choice of the smoothing parameter or knot mesh. Since it takes very little time for COBS to return the fitted curve for each set of chosen parameters, visual comparison and judgment can be performed interactively. The second option provides full automation. Users are not required to supply any smoothing parameter; COBS makes adaptive choices using information criteria similar to those in model selection.

The L_1 framework leads to linear programming (LP) formulations of the computational problems and allows efficient computation via standard linear programming techniques. The LP form makes it possible to naturally incorporate all the constraints discussed above. As far as we know, COBS is the only smoothing algorithm that can do this without substantial increase in computational costs.

Robust smoothing via L_1 methods was also investigated in Wang & Scott (1994). COBS facilitates more than just robust function estimation via conditional median estimation of the response given the covariate. It also provides computation of other conditional quantile functions which have gradually become an integral part of data analysis. See Koenker & Bassett (1978) for their pioneering work on regression quantiles.

In Section 2, we introduce linear and quadratic splines and describe how the L_1 minimization problem can be solved as a linear program. Section 3 discusses how COBS chooses the smoothing parameter or the knot mesh if full automation is desired. Section 4 describes some additional features of COBS and finally three illustrative examples are provided in Section 5. The underlying Fortran program and S-plus interface for COBS are available from www.econ.uiuc.edu/∼ng or www.stat.uiuc.edu/∼he/software.html. The S-plus code used for the examples in this paper are also available.

2 Constrained Smoothing

For a pair of bivariate random variables (X, Y), the τth conditional quantile function, $g_\tau(x)$, of Y given $X = x$ is a function of x such that

$$P\left(Y \leq g_\tau(x) \,|X = x\right) = \tau.$$

The conditional median function ($\tau = .5$) provides a measure of central tendency and can be used to describe the overall relationship between X and Y. When a more complete picture of the relationship is needed, the whole spectrum of conditional quantile functions can be examined.

Given n pairs of realizations $\{(x_i, y_i)\}_{i=1}^n$ with $a = x_0 < x_1 < \cdots < x_n < x_{n+1} = b$, some *smooth* function g and the *check function* $\rho_\tau(u) = 2\left[\tau - I(u < 0)\right] u = \left[1 + (2\tau - 1)\, sgn\,(u)\right] |u|$ with $I(\cdot)$ being the indicator function, we define "fidelity" to the data as

$$\text{“fidelity”} = \sum_{i=1}^{n} \rho_\tau(y_i - g(x_i)).$$

Koenker et al. (1994) introduced the τth L_p *quantile smoothing spline*, $\hat{g}_{\tau,L_p}(x)$, which is the solution to

$$\min_g \quad \text{“fidelity”} + \lambda \text{ “}L_p\,\text{roughness”} \tag{1}$$

as a nonparametric estimator for $g_\tau(x)$. The usual smoothing parameter λ controls the trade-off between fidelity to the data and roughness of the fit. The smoothing spline $\hat{g}_{\tau,L_p}(x)$ becomes an interpolating function as $\lambda \to 0$ and corresponds to a linear fit when $\lambda \to \infty$. Two versions of the roughness measure,

$$\text{“}L_1 \text{ roughness”} = V\left(g'\right) = \sum_{i=1}^{n-2} \left|g'\left(x_{i+1}^{+}\right) - g'\left(x_i^{+}\right)\right| \tag{2}$$

and

$$\text{“}L_\infty \text{ roughness”} = \| \, g'' \, \|_\infty = \max_x g''(x) \tag{3}$$

were suggested, where $V(.)$ denotes the total variation norm. They show that $\hat{g}_{\tau,L_1}(x)$ is a linear (second order) smoothing spline for the L_1 roughness penalty while $\hat{g}_{\tau,L_\infty}(x)$ can be approximated by a quadratic (third order) smoothing spline for the L_∞ roughness penalty.

In this section, we will concentrate on the special case of $\tau = .5$ so "fidelity" is measured by the L_1 norm,

$$\text{“fidelity”} = \sum_{i=1}^{n} \left|y_i - g\left(x_i\right)\right|.$$

The resulting linear and quadratic smoothing splines are two alternative estimators for the conditional median function.

It is well known that any mth order smoothing spline with simple knots at $x_1, \ldots, x_n$ has an equivalent B-spline representation on the same knot sequence. We, however, start with a more general knot mesh $T = \{t_i\}_{i=1}^{N+2m}$ with $t_1 = \cdots = t_m < t_{m+1} < \cdots < t_{N+m} < t_{N+m+1} = \cdots = t_{N+2m}$. The motivation for this generalization, pertaining to computational efficiency considerations, will be explained in Section 2.1. The B-spline representation, $s \in S_{m,T}$, of a smooth function becomes

$$s(x) = \sum_{j=1}^{N+m} a_j B_j(x)$$

where N is the number of internal knots, $B_j(x)$ are the normalized B-spline basis functions, a_j are the coefficients for the B-spline basis functions and $S_{m,T}$ is the space of polynomial splines of order m with mesh T. See De Boor (1978), Dierckx (1993) or Schumaker (1981) for more details. An elegant presentation of flexible smoothing with B-splines and penalties is provided by Eilers & Marx (1996).

2.1 Smoothing B-Splines

For the sake of expositional convenience, we assume in this subsection that the x_i are all distinct from one another. We use linear B-splines ($m = 2$) with $N = n - 2$ internal knots in the mesh $T = \{t_i\}_{i=1}^{N+2m}$ such that $t_1 = t_m = x_1, t_{m+1} = x_2, \ldots, t_{N+m} = x_{n-1}, t_{N+m+1} = t_{N+2m} = x_n$ for the optimization problem of the linear smoothing spline in (1) and (2). Now the objective function can be written as

$$\begin{aligned} \min_{\theta \in R^{N+m}} \quad & \sum_{i=1}^{n} \left| y_i - \sum_{j=1}^{N+m} a_j B_j(x_i) \right| \\ & + \lambda \sum_{i=1}^{N} \left| \sum_{j=1}^{N+m} a_j B_j'(t_{i+m}) - \sum_{j=1}^{N+m} a_j B_j'(t_{i+m-1}) \right| \end{aligned}$$

where $\theta = (a_1, \ldots, a_{N+m})$.

We can express the above in a more compact form as

$$\min_{\theta \in R^{N+m}} \sum_{i=1}^{n+N} |\tilde{y}_i - \tilde{x}_i \theta| \tag{4}$$

where

$$\tilde{y} = \begin{pmatrix} y \\ \mathbf{0} \end{pmatrix}$$

is an $(n+N)\times 1$ *pseudo response* vector,

$$\tilde{X}=\left[\begin{array}{c}\mathbf{B}\\ \lambda\mathbf{C}\end{array}\right]$$

is an $(n+N)\times(N+m)$ *pseudo design* matrix with

$$\mathbf{B}=\left[\begin{array}{ccc} B_1(x_1) & \cdots & B_{N+m}(x_1)\\ \vdots & \cdots & \vdots\\ B_1(x_n) & \cdots & B_{N+m}(x_n)\end{array}\right]$$

and

$$\mathbf{C}=\left[\begin{array}{ccc} B_1'(t_{m+1})-B_1'(t_m) & \cdots & B_{N+m}'(t_{m+1})-B_{N+m}'(t_m)\\ \vdots & \cdots & \vdots\\ B_1'(t_{N+m})-B_1'(t_{N+m-1}) & \cdots & B_{N+m}'(t_{N+m})-B_{N+m}'(t_{N+m-1})\end{array}\right]$$

The fitted curve, $\hat{m}_{\lambda,L_1}(x)=\sum_{j=1}^{N+m}\hat{a}_jB_j(x)$, is a *linear (median) smoothing B-spline.*

The objective function (4) can be solved by any efficient linear programming algorithm. To see this, rewrite (4) as

$$\min\left\{\mathbf{1}'(u+v)\,|\tilde{y}-\tilde{X}\theta=u-v,\left(u^{'},v^{'}\right)\in R_{+}^{2(n+N)}\right\}. \qquad (5)$$

A modification of Bartels & Conn (1980)'s non-simplex active-set algorithm for the quantile smoothing splines described in Ng (1996) and Koenker & Ng (1996) can be easily adapted for (5).

Similarly, using quadratic ($m=3$) B-splines with $N=n-2$ internal knots in the mesh $T=\{t_i\}_{i=1}^{N+2m}$ such that $t_1=t_2=t_m=x_1,t_4=x_2,\ldots,t_{N+m}=x_{n-1},t_{N+m+1}=t_{N+2m-1}=t_{N+2m}=x_n$, we rewrite (1) and (3) as

$$\min_{\theta\in R^{N+m}}\sum_{i=1}^{n}\left|y_i-\sum_{j=1}^{N+m}a_jB_j(x_i)\right|+\lambda\max_x\sum_{j=1}^{N+m}a_jB_j''(x)$$

where $\theta'=(a_1,\ldots,a_{N+m})$. This is equivalent to

$$\min_{\theta\in R^{N+m+1}}\sum_{i=1}^{n}\left|y_i-\sum_{j=1}^{N+m}a_jB_j(x_i)\right|+\lambda\sigma$$

$$s.t. \qquad -\sigma\leq\sum_{j=1}^{N+m}a_jB_j''(t_{i+m-1})\leq\sigma \text{ for } i=1,\ldots,N+1$$

where $\theta'=(a_1,\ldots,a_{N+m},\sigma)$. In a more compact form, we have

$$\min_{\theta\in R^{N+m+1}}\sum_{i=1}^{n+1}|\widetilde{y}_i-\widetilde{x}_i\theta| \qquad (6)$$

$$s.t. \qquad \widetilde{D}\theta = \left[\begin{array}{cc} \mathbf{D} & \mathbf{1} \\ -\mathbf{D} & \mathbf{1} \end{array} \right] \theta \geq \mathbf{0}$$

where

$$\tilde{y} = \left(\begin{array}{c} y \\ 0 \end{array} \right)$$

is an $(n+1)$ pseudo response vector,

$$\tilde{X} = \left[\begin{array}{cc} \mathbf{B} & \mathbf{0} \\ \mathbf{0} & \lambda \end{array} \right]$$

is an $(n+1) \times (N+m+1)$ pseudo design matrix and

$$\mathbf{D} = \left[\begin{array}{ccc} B_1''(t_m) & \cdots & B_{N+m}''(t_m) \\ \vdots & \ldots & \vdots \\ B_1''(t_{N+m}) & \cdots & B_{N+m}''(t_{N+m}) \end{array} \right].$$

The resulting fitted curve, $\hat{m}_{\lambda,L_\infty}(x) = \sum_{j=1}^{N+m} \hat{a}_j B_j(x)$, is a *quadratic (median) smoothing B-spline.*

The LP equivalence of (6) is

$$\min \left\{ \mathbf{1}'(u+v) \,|\tilde{y} - \tilde{X}\theta = u - v, \widetilde{D}\theta \geq \mathbf{0}, \left(u', v'\right) \in R_+^{2(n+N)} \right\}. \qquad (7)$$

The pseudo design matrices in (5) and (7) are both of the order $O\left(n^2\right)$. This will impose a huge burden on computational speed and memory space for large data sets. But this can be alleviated by approximating the smoothing splines using a smaller number of internal knots N and hence reducing the order of the pseudo design matrices to $O(nN)$. For example, we can use $T = \{t_i\}_{i=1}^{N+2m}$ with t_i chosen to be the N ($\ll n$) sample quantiles of the covariate x, see Section 3 for further details.

2.2 Imposing Additional Constraints

Due to the LP nature of the problems (5) and (7), many qualitative restrictions on the fitted curves can be incorporated easily by the addition of equality or inequality constraints as described below.

Monotonicity Constraints

For the linear spline $\hat{m}_{L_1}(x)$, the additional set of constraints needed is

$$\mathbf{H}\theta \geq \mathbf{0}$$

for increasing functions and

$$\mathbf{H}\theta \leq \mathbf{0}$$

for decreasing functions where

$$\mathbf{H}=\left[\begin{array}{ccc} B_1'(t_m) & \cdots & B_{N+m}'(t_m) \\ \vdots & \cdots & \vdots \\ B_1'(t_{N+m+1}) & \cdots & B_{N+m}'(t_{N+m+1}) \end{array}\right].$$

For the quadratic spline $\hat{m}_{L_\infty}(x)$, the extra set of $N+2$ constraints is

$$\left[\begin{array}{cc} \mathbf{H} & \mathbf{1} \end{array}\right]\theta \geq \mathbf{0}$$

for increasing functions and

$$\left[\begin{array}{cc} \mathbf{H} & \mathbf{1} \end{array}\right]\theta \leq \mathbf{0}$$

for decreasing functions.

Convexity Constraints
For $\hat{m}_{L_1}(x)$, we need the N constraints

$$\mathbf{C}\theta \geq \mathbf{0}.$$

For $\hat{m}_{L_\infty}(x)$, the additional set of $N+1$ constraints is

$$\left[\begin{array}{cc} \mathbf{D} & \mathbf{0} \end{array}\right]\theta \geq \mathbf{0}.$$

Concavity restriction can similarly be imposed with all the inequalities reversed.

Periodicity Constraints
A restriction of the form $g(x_1)=g(x_n)$ is useful for cyclical time series where x_1 and x_n are the first and last unique observed values in the time domain of a cycle, e.g. the first $(x_1=1)$ and last $(x_n=12)$ months of a year in monthly data. This can be achieved easily with the addition of the single equality constraint

$$\left[\tilde{X}_{(1)}-\tilde{X}_{(n)}\right]\theta=0$$

where $\tilde{X}_{(1)}$ and $\tilde{X}_{(n)}$ are the first and nth row of the pseudo design matrix $\tilde{X}$.

Pointwise Constraints
Pointwise constraints on the function and/or its derivatives can be directly imposed on the coefficients of the spline as illustrated in Section 4.3.

2.3 Regression B-splines

The computational burden can be ameliorated in a different way by dropping the penalty term totally; i.e. setting $\lambda=0$ in (1). This gives rise to the *(median) regression B-splines* of He & Shi (1994). Fidelity in regression B-splines is still measured the same way as in smoothing splines but roughness is controlled by the number of internal knots N rather than the smoothing parameter λ.

The *linear (median) regression B-spline*, $\hat{m}_{T,L_1}$, will solve

$$\begin{array}{lc} \min_{\theta \in R^{N+m}} & \sum(u+v) \\ s.t. & y - \tilde{X}\theta = u - v \\ & u \in R^n_+,\ v \in R^n_+ \end{array}$$

where

$$\tilde{X} = \mathbf{B}$$

is now an $n \times (N+m)$ pseudo design matrix with $m = 2$. The *quadratic (median) regression B-spline,* $\hat{m}_{T,L_\infty}$, solves the same minimization problem with $m = 3$.

The quantity $(N+m)$ plays the role of *effective dimensionality* of the fit. The two extreme fits correspond to $N = 0$, which yields the globally linear and quadratic B-spline fits for $m = 2$ and $m = 3$ respectively, while $N = n-2$ with $t_{i+m-1} = x_i$ for $i = 1, \ldots, n$, gives the interpolating fit.

3 Choice of Smoothing Parameter or Knots

The calling sequence of COBS is given in the Appendix. If a fully automated smoother is required, we must resolve the issue of choosing either the smoothing parameter λ for the smoothing splines or the knot mesh $T = \{t_i\}_{i=1}^{N+2m}$ in the case of regression B-splines. Asymptotically, the generalized cross-validation (GCV) criterion commonly used in least squares based smoothing splines is equivalent to the Akaike information criterion (AIC). AIC is similar to the Schwarz information criterion (SIC) for moderate sample sizes. For our L_1-type objective function, however, the projection based GCV can not be as directly motivated.

When the argument `lambda` is supplied with a negative value, COBS computes the smoothing spline with λ chosen to minimize a Schwarz-type information criterion used in Koenker et al. (1994), and He, Ng & Portnoy (1998). Denote $\hat{m}_{\lambda,L_1}$ or $\hat{m}_{\lambda,L_\infty}$ simply as $\hat{m}_\lambda$, our variant of SIC is defined as

$$SIC(\lambda) = \log(\frac{1}{n}\sum_i \rho_\tau(y_i - \hat{m}_\lambda(x_i))) + \frac{1}{2}p_\lambda \log(n)/n$$

where p_λ is the number of interpolated data points and serves as dimensionality measure of the fitted model. When $\tau = .5$, we may view the above SIC as the Gaussian likelihood based information criterion of Schwarz (1978) where the root mean square error is replaced by a robust alternative using the mean absolute residual as a measure of fidelity to the observed data.

From the LP nature of the objective functions, we know there are only finitely many distinct $\hat{m}_\lambda$ as λ varies over $(0, \infty)$. Parametric linear programming (PLP) as described in Ng (1996) is used to obtain all the possible distinct $\hat{m}_\lambda$ when λ decreases to zero from a value specified in the argument `lstart`.

The total number of distinct λ values grows with the sample size n. As a result, PLP will rapidly become the computational bottle-neck for moderately large n. From our experience, $\hat{m}_\lambda$ is not very sensitive to small perturbation in the λ values. In typical cases, there exist several very similar solutions corresponding to neighboring values of λ obtained from PLP. To speed things up, COBS allows users to skip some neighboring λ values via the argument `factor` by specifying 1 <`factor`< 4 while performing PLP. A bigger `factor` allows more neighboring λ's to be skipped. This is equivalent to using a coarser grid in the search for the optimal λ.

It is important to note that the first term of *SIC* becomes infinitely small if $\hat{m}_\lambda$ interpolates every single data point. As a result, the λ that minimizes *SIC* could be too small for unconstrained fits. Since *SIC* is meant to be used for model comparison with dimensionality not too close to n, COBS displays a warning message recommending the user examine the plot of *SIC* against λ when the chosen λ is near zero to see if the second minimizer of *SIC* will provide a more reasonable fit. COBS returns the necessary plotting information in the components `$pp.lambda` and `$sic`.

When `lambda` is provided with a positive number, it will be used as the value of the smoothing parameter. No efforts will be made to choose the optimal λ. This option allows the user to experiment with various fits of different smoothness.

The argument `knots` allows users to specify the location of the knots while `nknots` is used to control the number of knots. If `knots` is missing, a default set of `nknots` knots will be generated by one of the two methods specified by the `method` argument. The default `method` is '`quantile`', which uses `nknots` design points uniform in their percentile levels as the knot sequence. For example, if `nknots = 3`, the median of the covariate will be the single internal knot. The quartiles will be used if `nknots` = 5. If `method` = '`uniform`', uniformly spaced points between the smallest and largest design values will be used as the knot sequence. COBS will display an error message if there is no observation which falls between any pair of adjacent knots when the '`uniform`' option is chosen. If `nknots` is missing, a default value of 20 is assigned.

When `lambda` is set to zero, COBS computes the regression B-spline estimate. If both `knots` and `nknots` are provided and `nknots` equals the length of `knots`, COBS uses the supplied knot sequence without performing the knot selection procedure **Step 1** $-$ **3** below. This allows users to interactively experiment with various fits for their specified sets of knots. Otherwise, the knot selection procedure described below will be performed. If `knots` is missing, COBS will generate a default set of `nknots` (default to 6 if `nknots` is missing as well) knots by one of two methods specified through `method` as described above. If `knots` is not missing, COBS will use it to begin the following knot selection procedure:

Step 1: Choose the initial optimal number of internal knots N. Compute the regression B-splines for $N = 0, \ldots,$ (`nknots` -2) internal knots from `knots`. Denote $\hat{m}_{T,L_1}$ or $\hat{m}_{T,L_\infty}$ simply as $\hat{m}_T$, select N that corresponds to the smallest

$$AIC(T) = \log(\frac{1}{n}\sum_i \rho_\tau(y_i - \hat{m}_T(x_i))) + 2(N+m)/n$$

where T is the knots mesh, N is the number of internal knots in T, and m is the order of spline used.

Step 2: Perform stepwise knot deletion. Each of the internal knots is deleted sequentially to obtain a sequence of AIC values. The one whose deletion leads to the largest reduction in AIC is then slated for actual deletion. This process is repeated until no more existing knot can be removed.

Step 3: Perform stepwise knot addition. When the argument `knots.add` is set to `TRUE` , COBS takes the mid-point between every adjacent pair of existing knots as potential new candidate. If inclusion of any such point reduces the value of AIC, we choose to add the one which provides the largest reduction in AIC as long as there are observations between the knots. This process repeats until no more knot needs to be added.

We should note that in the current implementation, we do not cycle through **Step 2** and **3** repeatedly. That is, we do not go back to the knot deletion process after knot addition.

We use the constant $2/n$ in the second term of AIC as it appears to give the best overall results in our experiments with monotone and concave/convex functions. Should AIC undersmooth the data, users also have the option of using the SIC by substituting $\log(n)/n$ for $2/n$. This is done in COBS via assigning `ic='sic'` instead of the default setting of `ic='aic'`.

In constrained smoothing, the number of knots needed is typically small. The initial number of knots chosen by AIC in **Step 1** is often less than 4 when the sample size is not too large. However, if the chosen number is $N = 4$, we may want to investigate estimated fits with larger N values. COBS will remind the user of this through a warning message at the end of the computation. The user can then re-run the program with a larger value of `nknots` if `knots` is missing, or supply COBS with a longer `knots` sequence.

4 Other Features of COBS

In this section, we discuss additional issues related to the design and use of COBS.

4.1 Large Sample Problems

The amount of time needed to obtain an estimate may disappoint even the most patient users when a fully automated solution is chosen for a large data set. COBS seeks an approximate solution to the choice of smoothing when the sample size n exceeds 1000. This is achieved by using a subset of the original data during the selection of the smoothing parameter λ. The sub-sample size is chosen to be $n_* = 670 + \log(n)^3$ rounded to the nearest integer. Assuming that the true function is twice differentiable, we know that the optimal smoothing parameter is in the order of $n^{1/5}$, see Portnoy (1997). The SIC-based choice of λ from the subset can be adjusted by multiplying a factor of $(n/n_*)^{1/5}$, and then the whole data set is used in the final model fitting stage. As noted in the previous section, the storage space needed in the regression B-spline (with `lambda`$= 0$) is of the order much smaller than the smoothing B-spline. We, therefore, recommend using the smoothing B-spline with a small number of internal knots N or the regression B-spline when the sample size is big and automation is required.

4.2 Roughness Penalty

COBS uses the argument `degree` to determine the type of roughness penalty for the smoothing B-splines. The L_1 roughness penalty is selected by setting `degree` $= 1$ for linear spline fits and the L_∞ roughness penalty is chosen with `degree` $= 2$ for quadratic splines.

4.3 Pointwise Constraints

Four types of pointwise constraints can be imposed in COBS via the argument `pointwise`, whose value is a three-column matrix with each row representing one of the following conditions:

$$\begin{array}{lll} (0, x, y) & \text{for} & g(x) = y; \\ (1, x, y) & \text{for} & g(x) \geq y; \\ (-1, x, y) & \text{for} & g(x) \leq y; \\ (2, x, y) & \text{for} & g'(x) = y. \end{array}$$

Multiple constraints are allowed. COBS performs a feasibility check to ensure that they do not contradict one another.

Pointwise constraints are particularly useful in imposing boundary conditions. For example, if the response variable is weight or salary, it is useful to impose the $(1, 0, 0)$ constraint which corresponds to $g(0) \geq 0$. For monotonically increasing function, this will imply $g(x) \geq 0$ for all $x > 0$. See Example 2 for an implementation of such boundary restriction.

The constraints we have included in COBS are certainly not meant to be exhaustive. Other types of pointwise restrictions could be added as need arises.

4.4 Binary Choice Model

In a binary choice model where there are only two possible values of the response variable, the L_1 based COBS may not be appropriate. COBS will, therefore, print out a warning message recommending users pre-smooth the data with one of the S-plus smoothers like `ksmooth`, `loess` or `smooth.spline`. The pre-smoothed fitted values can then be passed on to COBS to incorporate further monotonicity or pointwise constraints.

4.5 More on Other Quantiles

Conditional quantile functions beyond the median may provide a more complete picture of the relationship between the response and the covariate. COBS provides estimates of the τth conditional quantile via the `tau` argument. We should point out that AIC and SIC criteria may not perform as well in choosing the smoothing parameters for extreme quantiles (τ close to 0 or 1) as for the median. User intervention is recommended in such situations.

It is also possible that the conditional quantile estimates may cross each other in the areas where data are sparse. One way to avoid this is to begin with some specific quantile of interest, say the median. Pointwise constraints as discussed above can then be imposed on the subsequent quantiles of interest to ensure that a subsequent quantile falls above or below the previously estimated quantile. Another practical approach is to compute the restricted regression quantiles proposed in He (1997).

4.6 Speeding Up

Recently, Portnoy & Koenker (1997) propose an improved interior point algorithm to solve an LP problem like ours. Their idea is to combine the recent advances in interior point method with a new statistical pre-processing approach so the algorithm can handle massive data sets at a speed comparable to that of least squares computation. Although the current version of COBS has not adopted this new method, it does appear possible that a substantial improvement in computational speed can be achieved for massive data sets in the future.

5 Illustrative Examples Using COBS in S-plus

To help readers familiarize themselves with the COBS approach, we provide three examples in this Section using COBS in S-splus. The S-plus codes used to produce our results are available at the web site provided at the end of Section 1. In Examples 2 and 3, we find it more difficult to modify other smoothers to satisfy the necessary pointwise constraints.

Example 1: We consider the annual average global surface temperature measured in degrees K (Hansen & Lebedeff 1987, Hansen & Lebedeff 1988). The data covers the period of 1880 to 1992. The temperatures presented in Figure 1 are temperature deviations.

The monotonicity constraint was used under a strong assumption of global warming; see Figure 1a. The automated *AIC* knot selection criterion picked $N = 4$ (the largest allowed by the default value of `nknots` = 6) internal knots for the 50th percentile curve. As discussed in Section 3, COBS printed a warning message so we increased `nknots` incrementally to 9. The final knots selected are located at (1880, 1908, 1936, 1964, 1978, 1992). These same knots were used for the 10th and 90th percentile curves.

The extreme temperature years in the top and bottom 10% after adjusting for the overall trend of global warming (if so believed) can be readily identified in Figure 1a. The hotter years are 1889, 1897$*$, 1900, 1901, 1915$*$, 1926, 1937$*$, 1938$*$, 1940, 1947$*$, 1953, 1980$*$, 1981 and 1990$*$ with those years followed by an '$*$' fall exactly on the 90th percentile curve. The colder years are 1884, 1887$*$, 1904, 1907$*$, 1917, 1918, 1950$*$, 1956, 1964, 1965, 1971$*$, 1976 and 1992$*$ again with those followed by an '$*$' lie exactly on the 10th percentile curve. After adjusting for the trend, 1987, 1988, and 1991 would not be considered extreme as they all fell below the top 10th percentile curve.

If the assumption of rising temperatures is dropped, the unconstrained version of the curves are presented in Figure 1b. They show a cooling period from 1936 to 1964. These unconstrained percentile curves are quite similar to the linear quantile smoothing splines presented in Koenker & Schorfheide (1994, p.401, Fig 3) except their tenth percentile is somewhat oversmooth as compared to ours.

The percentiles curves provide an ordering of data adjusted for the overall trend. They also suggest that variability of global temperature is rather stable over the last century. We have not attempted to correct for serial correlations in the data in all the fits. Our main objective here is to demonstrate how COBS can be applied to real data sets. Readers are encouraged to refer to Koenker & Schorfheide (1994) for a more careful treatment of possible autocorrelation in the model.

Example 2: The US Army Construction Engineers use flashing condition index (FCI) as one of several important roof condition measures. Roughly speaking, FCI shows what percentage of roof flashing is in good condition. We use records from 153 roof sections with EPDM base flashing from a number of U.S. Army bases and wish to study how FCI decreases over time. The ages of the roof sections vary between several weeks to fifteen years. Due to skewness of the FCI distribution, it is especially helpful in this case to compute the percentile curves instead of the mean and variance functions. The three quartiles corresponding to $\tau = 0.25, 0.50$ and 0.75 are computed for this example.

In addition to the obvious constraint of monotonicity, the engineers suggested that the majority of the new roofs at age 0 should have FCI at 100. Hence, we need a boundary constraint of $g(0) = 100$ for these quartiles. We choose to use the quadratic smoothing B-splines in COBS. To ensure that enough observations fall between adjacent knots, we use ten distinct ages (equally spaced in their percentile ranks) as knots.

For each τ, we obtained a plot of SIC versus the values of λ. Examination of these plots proved to be useful. The SIC plot for $\tau = 0.5$ is presented in Figure 2a. The global minimum occurs at $\lambda = 21.58$. The corresponding median smoothing spline is given in Figure 2b. In this example, a range of larger λ's yield similar SIC values and similar fitted curves mainly due to the simplicity in data structure when the monotonicity constraint is present. The second minimum of SIC occurs at 239.24, and the corresponding median smoothing spline is also plotted in Figure 2b as the dotted curve. When a large value of λ is an acceptable choice, it suggests that the resulting fit is close to globally quadratic and the roughness penalty is near zero.

We can see from Figure 2b that the top 25% of the EPDM roofs still remain in perfect condition after fifteen years. In fact, about 38% (58/153) of the responses stay at 100 for the 15-year period. Even the lower quartile shows a very slow rate of degradation after the eighth year.

Example 3: This example serves to illustrate the use of `'periodic'` constraint for cyclical data. The response variable is the daily average wind speed (in knots) recorded at the synoptic meteorological station in Dublin, Ireland from 1961 to 1978. There are altogether 6574 observations. The data was analyzed in detail in Haslett & Raftery (1989) and can be downloaded from *statlib*. Here, we use the quadratic smoothing B-spline with thirteen knots which correspond roughly to the beginning of all the twelve months of a year. The data is plotted in Figure 3a. For $\tau = .5$, the initial λ chosen by SIC reached the largest possible value allowed by the default setting of `lstart`. As recommended in the warning message of COBS, we re-fit the model to allow the parametric linear programming in λ to begin from a larger λ value. The final fits for $\tau = .1, .5$, and $.9$ using λ values automatically selected from the SIC criterion are given in Figure 3b. Notice that for each of the quantiles, we have required that the fitted values at the beginning and the end of a year are the same. However, we have not attempted to correct for any possible correlation in the data. Another point worth noting is that the upper percentile curve looks rather rough. Further research is needed to determine how the SIC criterion should be adjusted when τ is close to 0 or 1.

References

Bartels, R. & Conn, A. (1980), 'Linearly constrained discrete l_1 problems', *ACM Transaction on Mathematical Software* **6**, 594–608.

Cleveland, W. (1979), 'Robust locally weighted regression and smoothing scatterplots', *Journal of the American Statistical Association* **74**, 829–836.

De Boor, C. (1978), *A Practical Guide to Splines*, Vol. 27 of *Applied Mathematical Sciences*, Springer-Verlag, New York.

Delecroix, X., Simioni, M. & Thomas-Agnan, C. (1995), 'A shape constrained smoother: Simulation study', *Computational Statistics* **10**, 155–175.

Dierckx, P. (1993), *Curve and Surface Fitting with Splines*, Clarendon, Oxford.

Eilers, P. H. C. & Marx, B. D. (1996), 'Flexible smoothing with b-splines and penalties', *Statistical Science* **11**, 89–102.

Friedman, J. (1984), A variable span smoother, Technical Report 5, Laboratory for Computational Statistics, Department of Statistics, Stanford, University, California.

Green, P. J. & Silverman, B. W. (1994), *Nonparametric Regression and Generalized Linear Models: A Roughness Penalty Approach*, Chapman Hall, London.

Hansen, J. & Lebedeff, S. (1987), 'Global trends of measured surface air temperature', **92**, 13345–13372.

Hansen, J. & Lebedeff, S. (1988), 'Global surface air temperature: Update through 1987', *Geophys. Lett.* **15**, 323–326.

Härdle, W. (1990), *Applied Nonparametric Regression*, Cambridge University Press, New York.

Haslett, J. & Raftery, A. (1989), 'Space-time modelling with long-memory dependence: Assessing ireland's wind power resource', **38**, 1–21.

Hastie, T. & Tibshirani, R. (1990), *Generalized additive models*, Chapman & Hall.

Hawkins, D. M. (1994), 'Fitting monotonic polynomials to data', **9**, 233–237.

He, X. (1997), 'Quantiles without crossing', *American Statisticians* **51**, 186–192.

He, X. & Shi, P. (1994), 'Convergence rate of b-spline estimators of nonparametric conditional quantile functions', *Journal of Nonparametric Statistics* **3**, 299–308.

He, X. & Shi, P. (1998), 'Monotone b-spline smoothing', *Journal of the American Statistical Association*. Forthcoming.

He, X., Ng, P. & Portnoy, S. (1998), 'Bivariate quantile smoothing splines', *Journal of the Royal Statistical Society, Series B*. Forthcoming.

Koenker, R. & Bassett, Gilbert, J. (1978), 'Regression quantiles', *Econometrica* **46**, 33–50.

Koenker, R. & Ng, P. (1996), 'A remark on bartels and conn's linearly constrained l1 algorithm', *ACM Transaction on Mathematical Software* **22**, 493–495.

Koenker, R. & Schorfheide, F. (1994), 'Quantile spline models for global temperature change', *Climate Change* **28**, 395–404.

Koenker, R., Ng, P. & Portnoy, S. (1994), 'Quantile smoothing splines', *Biometrika* **81**, 673–680.

Ng, P. (1996), 'An algorithm for quantile smoothing splines', *Computational Statistics and Data Analysis* **22**, 99–118.

Portnoy, S. (1997), 'Local asymptotics for quantile smoothing splines', *Annals of Statistics* **25**, 414–434.

Portnoy, S. & Koenker, R. (1997), 'The gaussian hare and the laplacian tortoise: Computability of squared-error vs. absolute-error estimators (with discussion)', *Statistical Science* **12**, 279–296.

Ramsay, J. O. (1988), 'Monotone regression splines in action', *Statistical Science* **3**, 425–441.

Schumaker, L. (1981), *Spline Functions: Basic Theory*, Wiley & Son, New York.

Schwarz, G. (1978), 'Estimating the dimension of a model', **6**, 461–464.

Wahba, G. (1990), *Spline Models for Observational Data*, SIAM:PA.

Wang, F. T. & Scott, D. W. (1994), 'The l_1 method for robust nonparametric regression', *JASA* **89**, 65–76.

Watson, G. S. (1966), 'Smooth regression analysis', *Sankhya* **26**, 359–378.

Wright, I. W. & Wegman, E. J. (1980), 'Isotonic, convex and related splines', *The Annals of Statistics* **8**, 1023–1035.

Appendix

Following is the calling sequence for COBS.

cobs(x, y, constraint, z, minz = knots[1], maxz = knots[nknots], nz = 100, knots, nknots, method = 'quantile', degree = 2, tau = 0.5, lambda = 0, ic = 'aic', knots.add = F, pointwise, print.warn = T, print.mesg = T, coef = rep(0, nvar), w = rep(1, n), maxiter = 20*n, lstart = log(.Machine$single.xmax) ** 2, factor = 1)

ARGUMENTS

x vector of covariate.

y vector of response variable. It must have the same length (n) as **x**.

constraint 'increase', 'decrease', 'convex', 'concave', 'periodic' or 'none'.

OPTIONAL ARGUMENTS

z vector of grid points at which the fitted values are evaluated; default to an equally spaced grid with **nz** grid points between **minz** and **maxz**. If the fitted values at **x** are desired, use z = unique(x).

minz needed if **z** is not given; default to min(**x**) or the first knot if **knots** are given.

maxz needed if **z** is not given; default to max(**x**) or the last knot if **knots** are given.

nz number of grid points in **z** if **z** is not given; default to 100.

knots vector of locations of the knot mesh; if missing, **nknots** number of **knots** will be created using the specified **method** and automatic knot selection will be carried out for regression B-spline (lambda = 0); if not missing and length(**knots**) == **nknots**, the provided **knot** mesh will be used in the fit and no automatic knot selection will be performed; otherwise, automatic knots selection will be performed on the provided **knots**.

nknots maximum number of knots; default to 6 for regression B-spline, 20 for smoothing B-spline.

method method used to generate **nknots** number of **knots** when **knots** is not provided; 'quantile' (equally spaced in percentile levels) or 'uniform' (equally spaced in covariate); default to 'quantile'.

degree degree of the splines; 1 for linear spline and 2 for quadratic spline; default to 2.

tau desired quantile level; default to 0.5 (median).

lambda penalty parameter; **lambda** == 0: no penalty (regression B-spline); **lambda**>0: smoothing B-spline with the given **lambda**; **lambda**<0: smoothing B-spline with lambda chosen by a Schwarz-type information criterion.

ic information criterion used in knot deletion and addition for regression B-spline method when **lambda** == 0; 'aic' (Akaike-type) or 'sic' (Schwarz-type); default to 'aic'.

knots.add logical; an additional step of stepwise knot addition will be performed for regression B-spline if T; the default is F.

pointwise an optional three-column matrix with each row specifying one of the following constraints: (1,xi,yi) – fitted value at xi will be >= yi; (-1,xi,yi) – fitted value at xi will be <= yi; (0,xi,yi) – fitted value at xi will be = yi; (2,xi,yi) – derivative of the fitted function at xi will be yi.

print.warn logical flag for printing of warning messages; default to T; probably needs to be set to F if performing monte carlo simulation.

print.mesg logical flag for printing of intermediate messages; default to T; probably needs to be set to F if performing monte carlo simulation.

coef initial guess of the B-spline coefficients; default to a vector of zeros.

w vector of weights the same length as **x** (**y**) assigned to both **x** and **y**; default to uniform weights adding up to one; using normalized weights that add up to one will speed up computation.

maxiter upper bound of the number of iteration; default to 20*n.

lstart starting value for lambda when performing parametric programming in lambda if **lambda**<0; default to log(.Machine$single.xmax)**2.

factor determines how big a step to the next smaller lambda should be while performing parametric linear programming in lambda; default to one will give all unique lambda's; use of bigger **factor** (> 1& < 4) will save time for big problems.

VALUE

coef B-spline coefficients.

fit fitted value at **z**.

resid vector of residuals from the fit.

z as in input.

knots the final set of knots used in the computation.

ifl exit code: 1 – ok; 2 – problem is infeasible, check specification of the **pointwise** argument; 3 – **maxiter** is reached before finding a solution, either increase **maxiter** and restart the program with **coef** and **knots** set to the value upon previous exit or use a smaller **lstart** value when **lambda**<0 or use a smaller **lambda** value when **lambda**>0; 4 – program aborted, numerical difficulties due to ill-conditioning.

icyc number of cycles taken to achieve convergence.

k the effective dimensionality of the final fit.

lambda the penalty parameter used in the final fit.

pp.lambda vector of all unique lambda's obtained from parametric programming when **lambda** <0 on input.

sic vector of Schwarz information criteria evaluated at **pp.lambda**.

Figure 1a. Monotonically increasing linear regression B-splines for global temperature at $\tau = .1, .5, .9$.

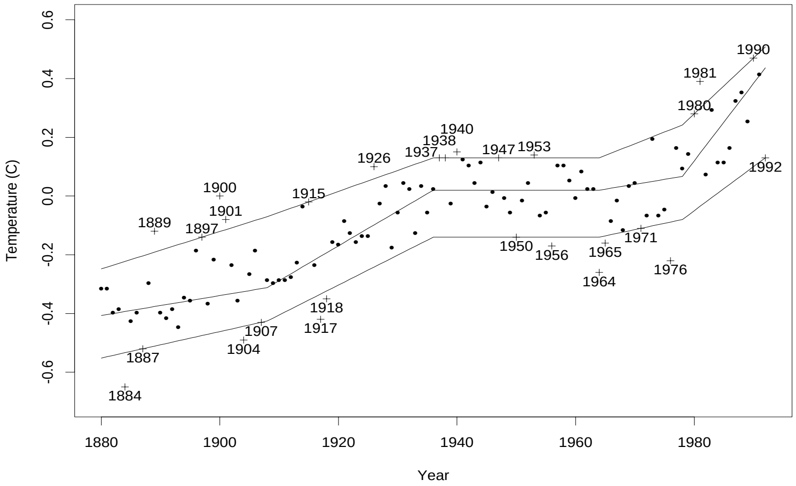


Figure 1b. Unconstrained linear regression B-splines for global temperature at $\tau = .1, .5, .9$.

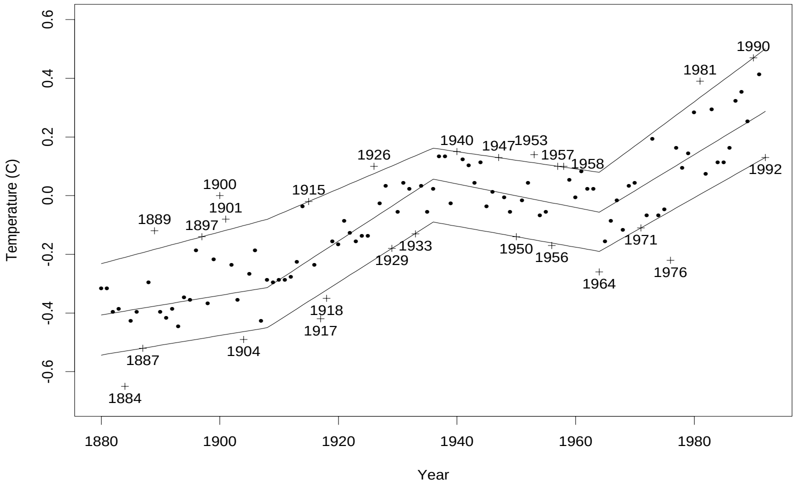


Figure 2a. *SIC* plot for the monotonically decreasing quadratic median smoothing B-spline for FCI degradation. A good choice of λ is at the global minimizer 21.57 but a range of larger values may also be considered in this example. The second smallest *SIC* value occurs at $\lambda = 239.24$ whose quadratic median smoothing B-spline fit is presented as the dotted curve in Figure 2b.

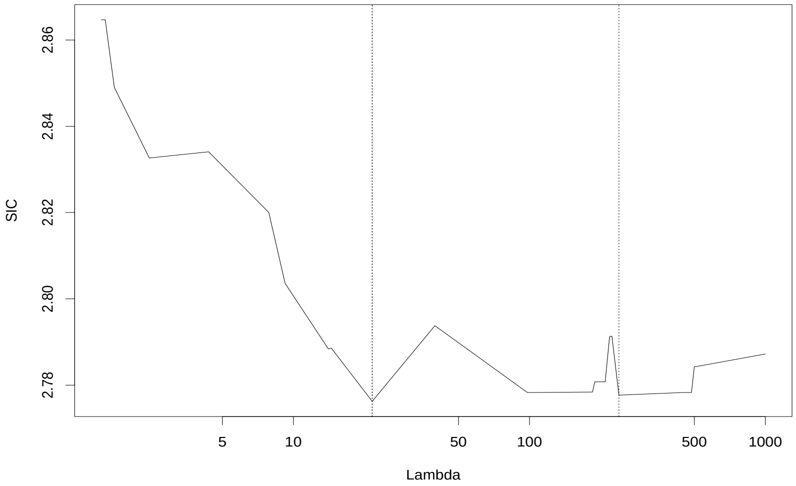


Figure 2b. Monotonically decreasing quadratic smoothing B-splines for FCI degradation at $\tau = .25, .5, (.5)$, and $.75$ with $\lambda = 57.62, 21.57, (239.24)$ and 10^8 respectively. A single point in the plot may represent multiple observations at the same location.

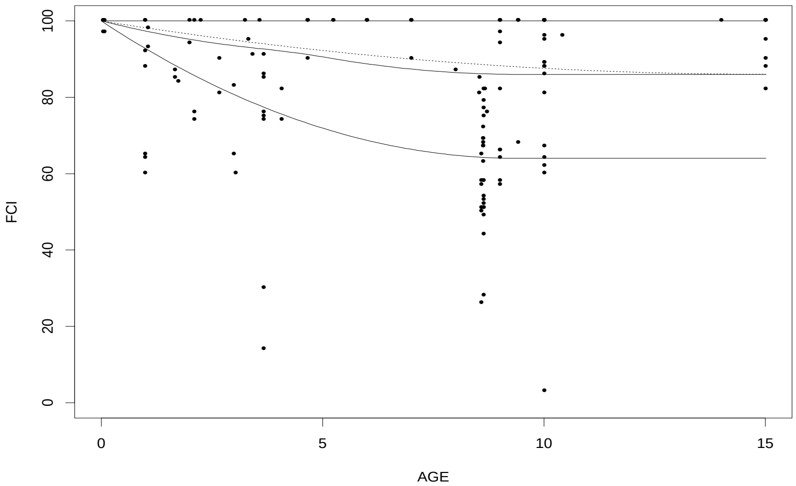


Figure 3a. Scatter plot of wind speed (in knots) in Dublin, Ireland.

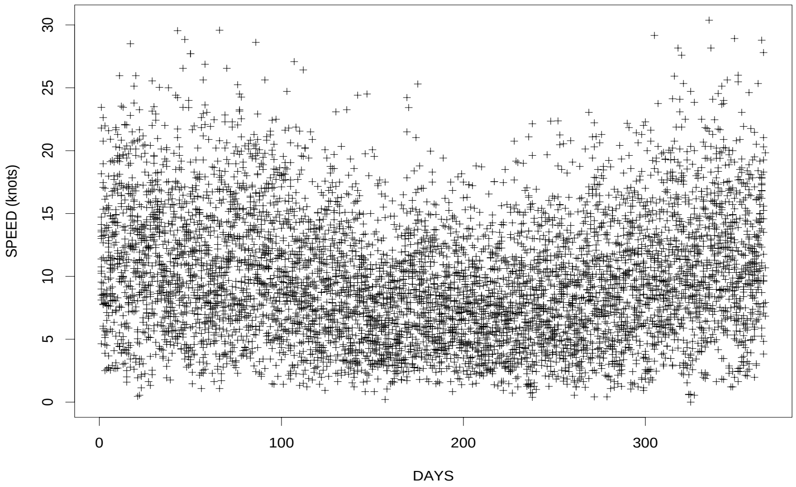
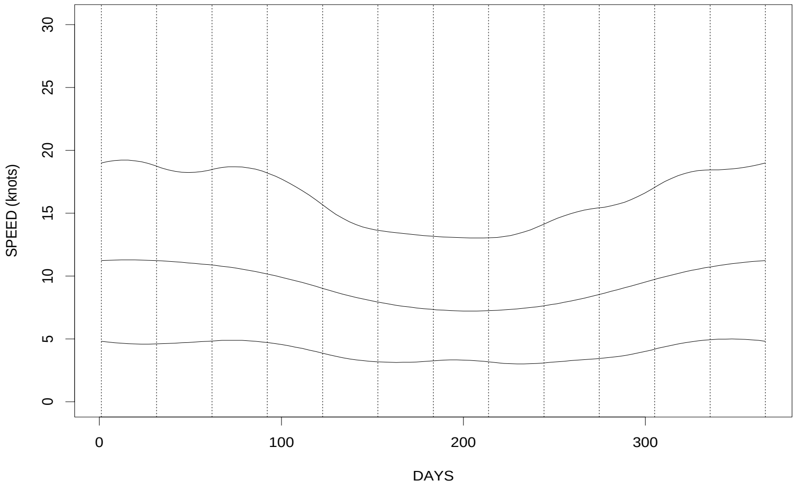


Figure 3b. The 'periodic' constrained quadratic smoothing B-spline fits for $\tau = .1, .5$, and .9. The smoothing parameters are $\lambda = 26782, 136589$, and 10367 respectively. The dotted lines indicate the location of the knots, which are the first days of each month.


SPEED (knots)
DAYS