A Fast and Efficient Implementation of Qualitatively Constrained Quantile Smoothing Splines

Pin Ng
College of Business Administration
Northern Arizona University
Flagstaff, Arizona 86011–5066
Email: Pin.Ng@nau.edu
Tel: (928) 523-8726
Fax: (928) 523-7331

Name of Conference: The 2005 International Conference on Algorithmic Mathematics and Computer Science

Abstract— **Exploiting the sparse structure of the design matrices involved in the Frisch-Newton method, we implement a fast and efficient algorithm to compute qualitatively constrained smoothing and regression splines for quantile regression. In a previous implementation (He and Ng, 1999), the linear program involved was solved using the non-simplex active set algorithm for quantile smoothing spines proposed by Ng (1996). The current implementation uses the Frisch-Newton algorithm described in Koenker and Ng (2005a, 2005b). It is a variant of the interior-point algorithm proposed by Portnoy and Koenker (1997) which has been shown to outperform the simplex method in many applications. The current implementation relies on the *R* package *SparseM* of Koenker and Ng (2003) which contains a collection of basic linear algebra routines for sparse matrices to exploit the sparse structure of the matrices involved in the linear program to further speed up computation and save memory usage. A small simulation illustrates the superior performance of the new implementation.**

Keywords: interior-point, simplex, linear program, quantile regression, smoothing spline

I. Introduction

Nonparametric regression has become widely popular in function estimation in recent years when the underlying relationship between the response and covariate is not believed to follow some parametric functional forms. Majority of nonparametric regression methods have been developed under the L_2 norm or the least squares framework where the resulting function, in general, provides an estimation of the conditional mean. Some robust versions that are based on the L_1 norm have also been developed. See Härdle (1990), Hastie & Tibshirani (1990), Green & Silverman (1994), Wang & Scott (1994), and He and Ng (1999) for additional references.

However, there are other aspects of the conditional relationship between the response and the covariate that will be useful and will shed new light on the underlying relationship. Relying on just the measures of central tendency will not reveal the additional information. Unless in the very naive homoscedastic situation where the variance of the response variable is the same across all the different regions of the covariate, quantile regression (Koenker and Bassett (1978)) provides a much fuller set of information on the dependence relationship over the whole spectrum of the response. The nonparametric quantile regression implemented here is based on the quantile smoothing splines proposed in Koenker, Ng and Portnoy (1994) and the quantile regression splines of He and Shi (1994), and He and Ng (1999).

In many function estimation applications, additional qualitative properties of the relationship are known a priori. For example in the construction of the growth chart, monotonicity is a desired property of the estimated function. In economics, the average cost function is known to be concave while the estimated production frontier is expected to be convex. For cyclical time-series data like river water level, atmospheric temperature or wind speed, one might want the first period of the fitted function to coincide with the fitted value at the last period of the year. In other situations, the fitted value might have to be above or below some specific threshold for the interpretation to be meaningful. In general, imposing these qualitative constraints is difficult in most nonparametric regression techniques. He and Ng (1999) and Mammen, Marron, Turlach and Wand (2001), however, provide a general framework in which these constraints can be incorporated easily into the estimation methods without imposing substantial computational or efficiency cost.

II. Constrained Smoothing

Since the current implementation is a more efficient and faster version of the constrained B-spline smoothing (COBS) proposed in He and Ng (1999), we will adopt most of the notations presented in that paper. For a pair of bivariate random variables (X, Y) and a scalar $\tau \in [0,1]$, the τth *conditional quantile function*, $g_\tau(x)$, of Y given $X = x$ is a function of x such that

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

Given n pairs of realizations $\{(x_i, y_i)\}_{i=1}^n$ of (X, Y) with $a = x_0 < x_1 < \cdots < x_n < x_{n+1} = b$, some *smooth* function g and the *check function* $\rho_\tau(r) = w_\tau(r)\,|r|$ where $w_\tau(r)$, defined as $[1 + (2\tau - 1)\,sgn\,(r)]$, is a weight function of the residual $r = y - g(x)$ and τ, Koenker, Ng and Portnoy (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 $g_\tau(x)$ where

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

and roughness is defined as either

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

or

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

As usual the smoothing parameter λ controls the trade-off between fidelity to the data and roughness of the fit. 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.

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. Using 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 B-spline representation, $s \in S_{m,T}$, of a smooth function, g, 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.

A. Quantile Smoothing B-Splines

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

$$\min_{\theta \in R^{N+m}} \sum_{i=1}^{n} \rho(y_i - s(x_i)) + \lambda \sum_{i=1}^{N} \left| s'(t_{i+m}) - s'(t_{i+m-1}) \right|$$

where $\theta = (a_1, \ldots, a_{N+m})^\top$ and $s'(t_{i+m}) = \sum_{j=1}^{N+m} a_j B_j'(t_{i+m})$.

Denoting $y = (y_1, \cdots, y_n)^\top$ and $w = (w_\tau(r_1), \cdots, w_\tau(r_n))^\top$, we can express the above in a more compact form as

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

where

$$\widetilde{w} = \begin{pmatrix} w \\ \mathbf{1} \end{pmatrix}$$

is an $(n+N) \times 1$ vector of weights,

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

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

$$\tilde{X} = \begin{bmatrix} \mathbf{B} \\ \lambda \mathbf{C} \end{bmatrix}$$

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

$$\mathbf{B} = \begin{bmatrix} B_1(x_1) & \cdots & B_{N+m}(x_1) \\ \vdots & \cdots & \vdots \\ B_1(x_n) & \cdots & B_{N+m}(x_n) \end{bmatrix}$$

and

$$\mathbf{C} = \begin{bmatrix} B_1'(t_{m+1}) - B_1'(t_m) & \cdots & B_{N+m}'(t_{m+1}) - B_{N+m}'(t_m) \\ \vdots & \cdots & \vdots \\ B_1'(t_{N+m}) - B_1'(t_{N+m-1}) & \cdots & B_{N+m}'(t_{N+m}) - B_{N+m}'(t_{N+m-1}) \end{bmatrix}$$

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

Rewrite (4) as

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

the objective function (4) can be solved by any efficient linear programming algorithm.

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

$$\min_{\theta \in R^{N+m}} \sum_{i=1}^{n} \rho_\tau(y_i - s(x_i)) + \lambda \max_x s''(x)$$

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

$$\min_{\theta \in R^{N+m+1}} \sum_{i=1}^{n} \rho_\tau(y_i - s(x_i)) + \lambda\sigma$$

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

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

$$\min_{\theta \in R^{N+m+1}} \sum_{i=1}^{n+1} \widetilde{w}_i \left| \widetilde{y}_i - \widetilde{x}_i \theta \right| \tag{6}$$

$$s.t. \qquad \widetilde{D}\theta = \begin{bmatrix} \mathbf{D} & \mathbf{1} \\ -\mathbf{D} & \mathbf{1} \end{bmatrix} \theta \geq \mathbf{0}$$

where

$$\widetilde{w} = \begin{pmatrix} w \\ \mathbf{1} \end{pmatrix}$$

is an $(n+1) \times 1$ vector of weights,

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

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

$$\tilde{X} = \begin{bmatrix} \mathbf{B} & \mathbf{0} \\ \mathbf{0}^\top & \lambda \end{bmatrix}$$

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

$$\mathbf{D} = \begin{bmatrix} B_1''(t_m) & \cdots & B_{N+m}''(t_m) \\ \vdots & \cdots & \vdots \\ B_1''(t_{N+m}) & \cdots & B_{N+m}''(t_{N+m}) \end{bmatrix}.$$

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

The LP equivalence of (6) is

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

B. Imposing Additional Constraints

Due to the LP nature of the problems in (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

The additional set of inequality constraints needed for the linear B-spline $\hat{m}_{L_1}(x)$ is

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

for increasing functions and

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

for decreasing functions where

$$\mathbf{H} = \begin{bmatrix} B_1'(t_m) & \cdots & B_{N+m}'(t_m) \\ \vdots & \cdots & \vdots \\ B_1'(t_{N+m+1}) & \cdots & B_{N+m}'(t_{N+m+1}) \end{bmatrix}.$$

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

$$\begin{bmatrix} \mathbf{H} & \mathbf{1} \end{bmatrix} \theta \geq \mathbf{0}$$

for $\hat{m}_{L_\infty}(x)$ to be increasing and

$$\begin{bmatrix} \mathbf{H} & \mathbf{1} \end{bmatrix} \theta \leq \mathbf{0}$$

for $\hat{m}_{L_\infty}(x)$ to be decreasing.

Convexity Constraints

For $\hat{m}_{L_1}(x)$ to be convex, we need the N inequality constraints

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

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

$$\begin{bmatrix} \mathbf{D} & \mathbf{0} \end{bmatrix} \theta \geq \mathbf{0}.$$

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

Periodicity Constraints

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, a restriction of the form $g(x_1) = g(x_n)$ is useful. For example in monthly data, the first month of a year has $x_1 = 1$ and the last month has $x_n = 12$. 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 also be directly imposed on the coefficients of the B-spline representations.

C. Quantile Regression B-splines

The pseudo design matrices $\widetilde{X}$ 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. In He and Ng (1999), this is 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 $O(nN)$. They suggest using $T = \{t_i\}_{i=1}^{N+2m}$ with t_i chosen to be the N ($\ll n$) sample quantiles of the covariate x. Another way to ameliorate the computational burden is to drop the penalty term totally; i.e. setting $\lambda = 0$ in (1). This gives rise to the *regression B-splines* of He and Shi (1994) when $\tau = 0.5$. Fidelity in quantile regression B-splines is still measured the same way as in smoothing B-splines, but roughness is controlled by the number of internal knots N rather than the smoothing parameter λ.

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

$$\min \left\{ \sum \tau \mathbf{1}^\top u + (1-\tau) \mathbf{1}^\top v | y - \tilde{X}\theta = u - v, \right. \\ \left. u \in R_+^n, \ v \in R_+^n \right\} \tag{8}$$

where

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

is now an $n \times (N+m)$ pseudo design matrix with $m = 2$. The *quadratic quantile 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 regression B-spline fits for $m = 2$ and $m = 3$ respectively, while $N = n-2$ with $t_{i+m-1} = x_i$ for $i = 1, \ldots, n$, results in the interpolating fit.

III. The Frisch-Newton Algorithm

The minimization problems in (5), (7) and (8) without the inequality constraints can be solved by any standard linear programming algorithm. Since its invention in 1947 by George B. Dantzig, the simplex method was basically the only effective method available for solving linear programs for nearly 40 years. In order to handle the inequality constraints imposed by the various qualitative constraints introduced in the previous section, He and Ng (1999) adapted a modification of Bartel and Conn's (1980) non-simplex active-set algorithm for the quantile smoothing splines detailed in Ng (1996) and Koenker and Ng (1996).

The publication of Karmarkar (1984) spawned a new class of algorithm for solving linear programs called "interior-point methods." Most interior-point software written since 1990 has been based on Mehrotra's (1992) predictor-corrector primal-dual algorithm. Portnoy and Koenker (1997) demonstrated evidence in accordance with the literature that the interior-point method is competitive with the simplex methods for moderate-sized problems, and it is superior to the simplex method for larger problems. This is particularly appealing to constrained quantile smoothing due to the rapid increase in dimension size of the pseudo design matrices as the sample size increases. Portnoy and Koenker (1997) coined their primal-dual interior-point implementation for quantile regression a Frisch-Newton algorithm because Frisch (1955) was a pioneering advocate of the log-barrier method, from which the interior-point approach had evolved.

To enable the incorporation of inequality constraints, Koenker and Ng (2005a) extended the Frisch-Newton algorithm of Portnoy and Koenker (1997) and derived the necessary linear algebra associated with the additional inequality constraints. Two distinct versions of the algorithm have been written in Fortran and linked to *R*, Ihaka and Gentleman (1996), which is an open source dialect of the statistical language *S* developed by Chambers (1998). One employs standard (dense) linear algebra routines from LAPACK while the other uses more specialized sparse linear algebra available in the *R* package *SparseM* of Koenker and Ng (2003) to improve performance for problems having a high proportion of zeros in the matrices involved. The latter formulation which is discussed in much more detail in Koenker and Ng (2005b) is particularly well-suited for the constrained quantile smoothing.

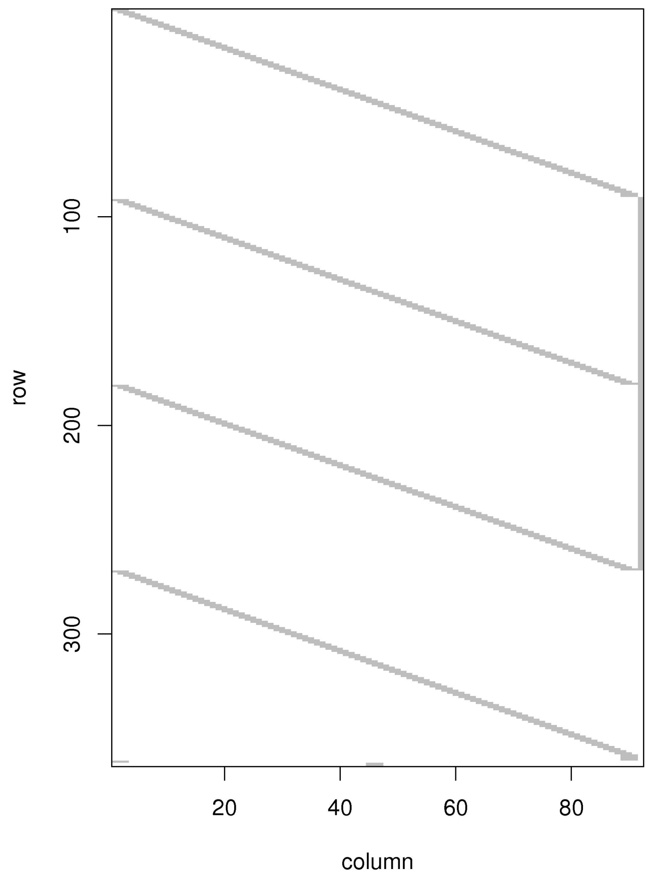

Fig. 1. The structure of a typical pseudo design matrix of COBS with $n = 100$. The grey areas represent the locations of the non-zero entries of the matrix while the white areas are where the zero entries are.

IV. Sparse Linear Algebra via SparseM

The Frisch-Newton algorithm is an iterative method and Koenker and Ng's (2005a) implementation consists of three components at each iteration: (i) an affine-scaling "predictor" direction is first computed, (ii) separate primal and dual step lengths are then obtained and the barrier parameter μ is updated so that , (iii) a "corrected" direction can be computed and a step taken along the corrected direction. Most of the computational effort of the Frisch-Newton algorithm is expanded on the solution of the linear system involving the very sparse pseudo design matrix in the affine-scaling and corrector steps. A typical pseudo design matrix of the constrained quantile smoothing B-splines with each unique x_i as a knot is presented in Figure 1 while Figure 2 shows the symmetric matrix involved in the linear system. The extremely sparse structure of the matrices is obvious. Noticing that the matrices involved are extremely sparse and also that the

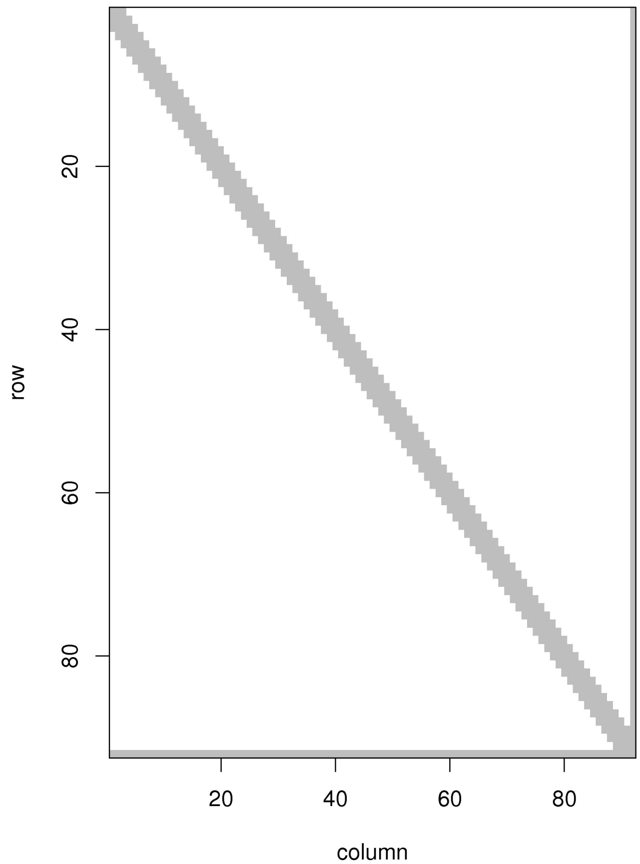

Fig. 2. The structure of a typical symmetric matrix with $n = 100$ that needs to be factorized in the affine-scaling step of the Frisch-Newton algorithm.

Cholesky factorization of the symmetric matrix has to be performed only once but can be used for both the affine-scaling and the corrector steps in each iteration, Koenker and Ng (2005b) provide an efficient implementation using the basic linear algebra routines specially designed for sparse matrices available from *SparseM*. They show that the factor of speed improvement ranges from roughly 36 at the sample size of 64 to approximately 850 at the sample size of 1024. While storage requirements of the dense matrix version increase quadratically, the sparse matrix implementation exhibits a linear increase in memory storage requirement. This enables problems with potentially massive-scale data to be solved on computers with limited memory capacity. The version of the Frisch-Newton algorithm that uses the specialized sparse linear algebra is the one that is implemented in this paper.

V. Simulation

To assess the efficiency gain of the Frisch-Newton algorithm tailored specifically for the sparse matrices when implemented in COBS, we perform a small simulation using the following test function:

$$\begin{aligned} y_i = \Phi(2x_i) + u_i \text{ for } i = 1, \cdots, n \\ s.t. \quad g(\min(x)) > 0 \\ g(\max(x)) < 1 \\ g(0) = 0.5 \\ g'(x) > 0 \end{aligned}$$

where $\Phi(\cdot)$ is the distribution function of a standardized normal random variable and u is drawn independently from a normal distribution with a mean of 0 and a standard deviation of 0.1. The median ($\tau = 0.5$) smoothing B-spline $\hat{m}_{\lambda, L_1}(x)$ is used to estimate $\Phi(\cdot)$. Since it takes a relatively long time for the non-simplex active set implementation of COBS to complete a single smoothing B-splines estimate with the full set of knots when the sample size increases beyond 500, we only perform 50 replications in the simulation for $n = \exp(4 \text{ to } 6.5 \text{ by } 0.5)$.

The median execution time, in seconds, of both the non-simplex active set implementation (COBS) and the sparse Frisch-Newton implementation (SCOBS) is reported in Figure 3. Reported in the legend are the least-squares estimated intercept and slope coefficients from regressing the log of execution time on the log of sample size. The estimated slope coefficient suggests that execution time grows exponentially at the rate of 3.71 for COBS and only 1.18 for SCOBS. At the very small sample size of 50, COBS is about twice as fast as SCOBS due to the heavier overhead in constructing the sparse matrix storage structure needed in *SparseM* in the SCOBS implementation. This overhead rapidly becomes insignificant as the sample size increases. At a sample size of $n = 1000$, SCOBS is already about 1000 times faster than COBS. Applications with a large number of observations are not uncommon nowadays. Some examples are the genomic microarray database and data collected on network traffic, etc. With a sample size of $n = 10,000$, the projected efficiency gain of SCOBS compared to COBS is a factor of around $350,000$. The enormous reduction in computational time and memory requirements open up a whole new arena for many potential applications that involve massive-scale data sets.

VI. References

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

Chambers, J. M. (1998). *Programming With Data: a Guide to the S Language*. Springer-Verlag.

Frisch, R. (1955). 'The logarithmic potential method of convex programming', Technical Report, University Institute of Economics, Oslo, Norway.

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

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

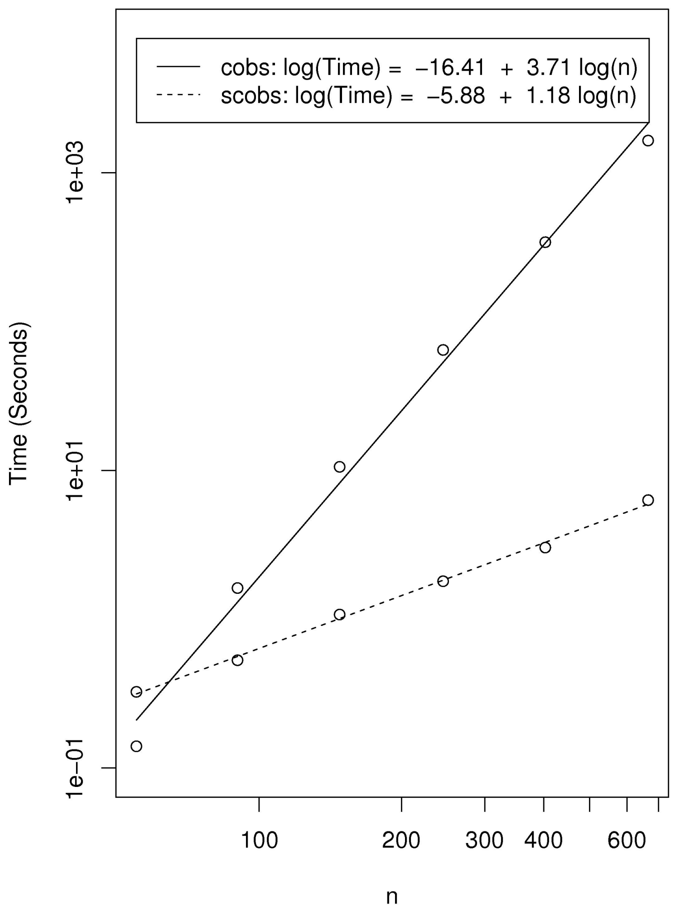

Fig. 3. Median execution time (in seconds) of COBS and SCOBS. Both time and sample sizes are in log-scale.

Hastie, T. and R. Tibshirani (1990). *Generalized Additive Models*, Chapman and Hall.

He, X. and P. Ng (1999). 'COBS: Qualitatively constrained smoothing via linear programming', *Computational Statistics*, 14, 315-337.

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

Ihaka, R., and R. Gentleman (1996). 'R: A Language for Data Analysis and Graphics', *Journal of Computational and Graphical Statistics*, 5(3), 299-314.

Karmarkar, N. (1984). 'A new polynomial time algorithm for linear programming', *Combinatorica*, 4, 373-395.

Koenker, R. and G. Bassett (1978). 'Regression quantiles', *Econometrica*, 46, 33-50.

Koenker, R. and P. Ng (1996). 'A remark on Bartels and Conn's linearly constrained l_1 algorithm', *ACM Transaction on Mathematical Software*, 22, 493-495.

Koenker, R. and P. Ng (2003). 'SparseM: a sparse matrix package for R', *Journal of Statistical Software*, 8, 2003.

Koenker, R. and P. Ng (2005a). 'Inequality constrained quantile regression', working paper.

Koenker, R. and P. Ng (2005b). 'A Frisch-Newton algorithm for sparse quantile regression', *Acta Mathematicae Applicatae Sinica*, forthcoming.

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

Mehrotra, S. (1992). 'On the implementation of a primal-dual interior point method', *SIAM Journal on Optimization,* 2, 575-601.

Mammen, E., J. S. Marron, B. A. Turlach and M. P. Wand (2001). 'A general projection framework for constrained smoothing', *Statistical Science*, 16, 232-248.

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

Portnoy, S. and R. Koenker (1997). 'The Gaussian hare and the Laplacian tortoise: computability of squared-error vs absolute error estimators', (with discussions), *Statistical Science*, 12, 279-300.

Wang, F. T. and D. W. Scott (1994). 'The l_1 method for robust nonparametric regression', *Journal of the American Statistical Association*, 89, 65-76.