Nonparametric regression subject to a given number of local extreme value

Ali Majidi and Laurie Davies

Department of Mathematics and Computer Science, University of Essen, Germany.

{ali.majidi,laurie.davies}@stat-math.uni-essen.de

Abstract

We consider the problem of nonparametric regression. The aim is to get a smooth function which represents the dataset and has a reasonable number of extreme values. An iterative method, the QSOR method is introduced. Problems with the slow convergence of the method are reduced using multigrid techniques.

1 Introduction

Given a dataset $\{y(t_i), i = 1, \ldots, n\}$ which we denote by y, we look for a decomposition

$$y(t_i) = f(t_i) + r(t_i), (t_i = i/n, i = 1, \ldots, n)$$

where f is a simple function and the $\{r(t_i), (i = 1, \ldots, n)\}$ are the resulting residuals which approximate white noise. We use two different concepts of simplicity. The first is the number of local extreme values. The second is the smoothness of the function as measured by the standard smoothness functional

$$S(f) := \int_0^1 (f^{(2)}(t))^2 dt,$$

where $f^{(2)}$ is the second derivative of f. The number of local extremes is taken to have priority over smoothness. The number of local extremes and their locations are determined by the taut string method developed in [3]. This is described briefly in the next section. The residuals are required to look like white noise in the sense that the means over certain dyadic intervals are required to lie within given bounds [3]. The multiresolution coefficients for $(n = 2^\nu)$ are defined by: $w_{ij} := 2^{(-i/2)} \sum_{k=j2^i+1}^{(j+1)2^i} r(t_k), (i = 0, \ldots, \nu), (j = 0, \ldots, 2^{(\nu-i)} - 1)$. The multiresolution condition now requires that $-c_n \leq w_{ij} \leq c_n$, where c_n represents some form of thresholding. The default value of c_n which we use is $c_n = \sigma_n \sqrt{2.5 \log(n)}$ where $\sigma_n = 1.482 \cdot \text{median}(|y_2 - y_1|, \ldots, |y_n - y_{n-1}|)/\sqrt{2}$.

Supported by SFB 475, University of Dortmund

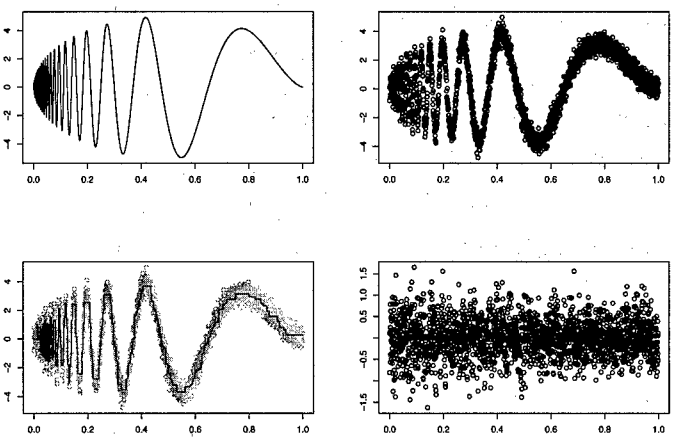

FIG. 1. The top-left caption shows the original doppler function and the top-right caption shows the noisy version. The bottom-left caption shows the result of the taut string algorithm with the resulting residuals being shown in the bottom-right caption.

2 Taut string

A short description of the taut string method is as follows. We write $f = (f_1, \ldots, f_n)^T := (f(t_1), \ldots, f(t_n))^T \in \mathbf{R}^n$ and denote the cumulative sums of y and f by Y and F respectively, $Y_i = \sum_{j=1}^{i} y_j$, $F_i = \sum_{j=1}^{i} f_j, (i = 0, \ldots, n)$, with $Y_0 = F_0 = 0$. We specify bounds defined by $\lambda = (\lambda_1, \ldots, \lambda_n)^T \in \mathbf{R}_+^n$ and consider the tube

$$\{G : |Y - G| \leq \lambda\}. \tag{2.1}$$

The taut string $V(\lambda)$ is now the function defined by a taut string attatched to the points $(0, Y_0)$ and (n, Y_n) and constrained to lie within the tube (2.1). It can be shown that the taut string minimizes the number of extreme values of the functions g whose cumulative sums G lie within the tube. The taut string is continuous and piecewise linear. Its derivative $v(\lambda)$ is taken as a candidate regression functions. The vector λ is determined in a data dependent manner by the requirement that the residuals associated with $v(\lambda)$ $\{r(\lambda)_i = y_i - v(\lambda)_i, i = 1, \ldots, n\}$ satisfy the multiresolution condition. If such a condition fails on an interval then the λ-values associated with that interval are reduced in size. An application of the taut string method to the doppler data of Donoho and Johnstone (see e.g. [4]) is shown in Figure 1. The function is defined by $f(t) = 21\sqrt{(v(1-t))}\sin\left(2\pi\frac{1+0.05}{t+0.05}\right)$. The derivative $v(\lambda)$ is piecewise constant as may be seen from Figure 1. The function $v(\lambda)$ determines the number of local extremes. We take the midpoints of the intervals associated with a local extremes as the locations of the local extremes for the smoothing algorithm.

3 The smoothing problem

We make the smoothing problem precise as follows. The number, locations and type of extreme values are taken from the taut string as explained in the last section. We further require the function f to lie in the tube determined by the taut string. This is to prevent the smoothing procedure from moving too far from the data. These restrictions may be described in the form

$$Af \geq b \tag{3.1}$$

for an appropriate matrix A and vector b. This leads to the following problem:

$$\text{minimize } \sum_{i=1}^{n}(f_{i+1} - 2f_i + f_{i-1})^2 \text{ subject to (3.1)},$$

or equivalently

$$\text{minimize } F^T Q_3 F \text{ subject to (3.1)},$$

for some quadratic form Q_3. We denote this latter quadratic programming problem by QP3. Clearly the matrix associated with the quadratic form $\sum_{i=1}^{n}(f_{i+1} - 2f_i + f_{i-1})^2$ is singular. Nevertheless the solution of QP3 may be unique. We have the following theorem.

Theorem 3.1 *Let $V(\lambda)$ be the result of the taut string method. Assume that $V(\lambda)$ has one extreme value. We define the bounds L, U by*

$$L := Y - \lambda, U := Y + \lambda.$$

Let $\hat{F}_1, \hat{F}_2$ be two solutions of the corresponding quadratic program. Additionally let $\hat{F}_1$ touch three bounds alternately

$$\textit{(i.e. } U_{i_1}, L_{i_2}, U_{i_3} \textit{ or } L_{i_1}, U_{i_2}, L_{i_3}, (i_1 < i_2 < i_3) \textit{ are active).}$$

Then

$$\hat{F}_1 = \hat{F}_2.$$

We call a problem with a unique solution a *nondegenerate* problem. From now on we assume that our problem is nondegenerate.

3.1 Quadratic programming

There are many algorithms which solve quadratic programming problems directly. Unfortunately most of them are expensive in terms of memory requirements and are not feasible for data sets of the order say $n = 8196$. To overcome this we look for iterative methods which converge to the solution. Gradient projection methods (e.g. as defined in [8], [2] or [9]) are not appropriate for this purpose as the monotonicity constraints make the projection into the feasible set too expensive. Instead we use a modified version of the QSOR (quasi successive over relaxation) method developed by Metzner in [7]. QSOR is a very cheap iteration and converges to the solution of QP3. Unfortunately the convergence is very slow on sections where the solution is smooth. To overcome this we use multigrid methods which have to be adapted to our requirements.

4 QSOR

The QSOR algorithm is an iterative method which produces a feasible sequence $\{F^k\}_{k=0}^{\infty}$ converging towards the solution of QP3. For simplicity, we describe the iteration only for a convexity interval. Let $F^0 \in \mathbf{R}^n$ be an arbitrary feasible vector. The obvious candidate is the derivative of the taut string. Let $Q = Q_3$ and $\omega \in (0, 2)$. The following defines a QSOR iteration.

- While convergence not achieved
- $F = F^k$
  $i = 1$
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qf)_i, \tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, L_i\}, \tilde{U}_i = U_i, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
  $i = 2$
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qz)_{ii}, \tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, L_i\}$
  $\tilde{U}_i = \min\{(F_{i+1} + F_{i-1})/2, U_i\}, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
- for (i in 3:(n-2)){
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qz)_i, \tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, 2F_{i-1} - F_{i-2}, L_i\}$
  $\tilde{U}_i = \min\{(F_{i+1} + F_{i-1})/2, U_i\}, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
  if(i active) mark i
  }
  }
  $i = n$
  $\tilde{F}_i = F_2 - \frac{\omega}{Q_{ii}}(Qz)_{ii}, \tilde{L}_i = \max\{2F_{i-1} - F_{i-2}, L_i\}, \tilde{U}_i = U_i, \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
  $i = 1$
  $\tilde{F}_i = F_i - \frac{\omega}{Q_{ii}}(Qz)_i$
  $\tilde{L}_i = \max\{2F_{i+1} - F_{i+2}, L_i\}$
  $\tilde{U}_i = U_i \hat{F}_i = \mathrm{med}\{\tilde{L}_i, \tilde{U}_i, \tilde{F}_i\}$
- correct the active intervals:
  * Let $[F_\nu, F_{\nu+k}]$ be an active Interval: $F_i = F_\nu + \frac{t_j - t_\nu}{t_{\nu+k} - t_\nu}(F_{\nu+k} - F_\nu)$. Denoting the i-th unit vector in $\mathbb{R}^n$ by e_i and a, b defined by

$$b = \frac{\sum_{i=\nu}^{\nu+k}(Qz)_i}{\sum_{i=\nu}^{\nu+k}\sum_{j=\nu}^{\nu+k} Q_{ij}}, \quad a = \frac{\sum_{i=\nu}^{\nu+k} t_i (Qz)_i}{\sum_{i=\nu}^{\nu+k} t_i \sum_{j=\nu}^{\nu+k} t_j Q_{ij}}$$

  set $F_j^k := \hat{F}_j - \hat{\alpha}(at_j + b)$ with
- $F_i^k = \hat{F}_i$ for all i in other intervals

Theorem 4.1 *(convergence) Let $(F^k)_{k=0}^{\infty}$ be the sequence in $\mathbb{R}^n$ produced by the QSOR algorithm and let the problem QP3 be nondegenerate. Then*

- *$(F^k)_{k=0}^{\infty}$ converges in $\mathbb{R}^n$.*
- *$F^* := \lim_{k\to\infty} F^k$ is the solution of QP3.*

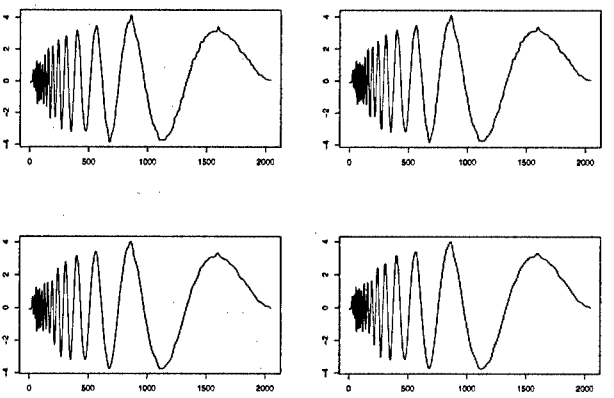

FIG. 2. The captions top-left, top-right, bottom-left, bottom-right show the result of the QSOR iteration for the doppler data ($n = 2048$) after $1000, 5000, 10000, 20000$ steps respectively.

The slowness of the convergence can be seen by the fact that the doppler data of Figure 1 required two million iterations before a satisfactory solution was obtained. This is shown in Figure 2. After a small number of iterations the solution does not change any more on the "left side" where the function oscilates rapidly. After 1000 iterations of QSOR (which is fast because one QSOR step is very cheap!) the solution looks very smooth except for a few "buckles" on the "right side" of the solution. The method needs many iterations (up to two million) to reach an adequate smoothness. The slowness of the convergence is known from the original SOR method for solving linear equations. In the standard case of solving linear equations multigrid methods can be used to speed up the rate of convergence. We now apply this idea to solving the problem QP3.

5 Multigrid QSOR

Multigrid techniques are general techniques to speed up iterative methods which indeed have other good properties. The ideas are given for example in [1] or [5]. We will give here a short description of the multgrid idea in our case. First some notation. Given a *grid* $G = G_f = (t_1, \ldots, t_n)$ we define the *coarse grid* $G_c = (t_1, t_{i_2}, \ldots, t_{i_{m-1}}, t_n), i_1 = 1, i_m = n$ with $i_j \in \{1, \ldots, n\}$. We define the projection *down* $I_c x = (F_1, F_{i_2}, \ldots, F_{i_{m-1}}, F_n)^T$ and the projection *up* $I^c x = y$ where $y_l = F_l$ $(l \in \{i_j | j = 1, \ldots, m\})$, and by linear interpolation elsewhere, i.e.,

$$y_l = \frac{F_{i_j} - F_{i_{j-1}}}{t_{i_j} - t_{i_{j-1}}} \quad (i_{j-1} < l < i_j).$$

We define now the *multigrid QSOR* with only two grids, i.e., of level two. The general case of level $\nu \in \mathbb{N}$ is defined similarly. Let $QSOR(G, A, b, \mu, x)$ denote the result of the

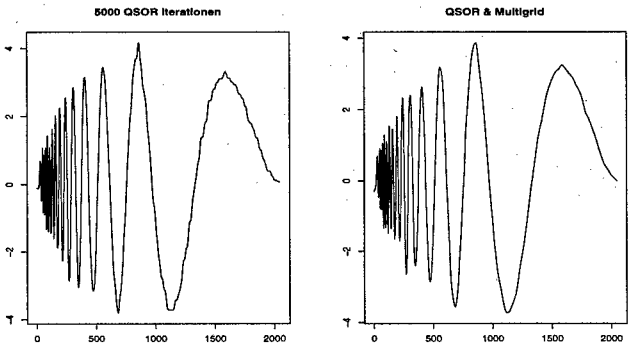

FIG. 3. The left figure shows the result of QSOR after 5000 iterations. The right figure shows the result of (1000) multigrid QSOR with one coarsing step (i.e. the right figure is "cheaper" than 2000 QSOR streps).

QSOR method applied to the problem on the grid G after μ iterations on the Grid G with starting vector x and constraints defined by A, b. Additionally let F^k be given.

- **Multigrid QSOR**
  - while precision not achieved
    - $\hat{F} = QSOR(G, A, b, \mu, F^k)$
    - $\tilde{F} = I^c QSOR(G_c, A_c, b_c, \mu, I_c \hat{F})$
    - $F^{k+1} = QSOR(G, A, b, \mu, \tilde{F})$
    - $k \leftarrow k + 1$

where A_c, b_c are the corresponding constraints for the coarser grid. The question is now how to define the *projection* of the constraints. One can think of an example where the canonical projection of bounds like G_c can fail. This happens for example if *strong* constraints (e.g. tight bounds) are not on the coarse grid. To overcome this problem one has to think of a method of defining the problem QP3 on the coarser grid in a reasonable way. One way to handle this problem is to define $L_{i_j} := \max\{L_k | i_{j-1} < k < i_{j+1}\}$ and "min" for the upper bounds. Special cases have to be treated but we do not go into details here. A coarser grid means that the QSOR step on this grid converges much faster. On the other hand the approximation of the solution gets worse by coarsening the grid. In our case (see Figure 4) we have $n = 2048$. The coarsest grid was made by taking every eighth gridpoint. We iterated until there was no recognizable improvement.

6 Proofs

Proof of Theorem 3.1: We set $D = \hat{F}_2 - \hat{F}_1$. One simply verifies that D has to be a line, i.e., there are $a, b \in \mathbf{R}$ such that $D_i = at_i + b$. Touching three bounds alternately means that D changes its sign at least two times which leads to $D = 0$. □

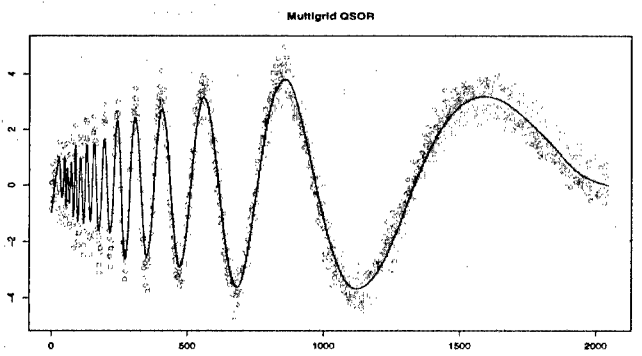

FIG. 4. Multigrid QSOR applied to the doppler data with $n = 2048$. The figure took less than 6 seconds comparing to three hours without multigrid on the same computer.

Proof of Theorem 4.1: We set $Q = Q_3$. We have to show:

1) $(S_3(F^k))_{k\in\mathbb{N}_0}$ decreases;
2) $(F^k)_{k\in\mathbb{N}}$ is feasible;
3) If F^s is a stationary point of QSOR, then F^s minimizes S_3 in the feasible set.

- our feasible set is compact, so the sequence has a convergent subsequence,
- a limit of a subsequence of $(F^k)_{k=1}^{\infty}$ is a stationary point of QSOR,
- the problem has only one solution.

To the first point, we only remark that a, b as defined in the algorithm, are the minimizers of the term:

$$\left(z - \left(x\sum_{i=\nu}^{\nu+k} t_i e_i + y\sum_{i=\nu}^{\nu+k} t_i e_i\right)\right)^T Q \left(z - \left(x\sum_{i=\nu}^{\nu+k} t_i e_i + y\sum_{i=\nu}^{\nu+k} t_i e_i\right)\right).$$

The others are treated as in [6]. The second point is clear, because by the definition we start with a feasible vector and we retain the feasibility in every single step. It remains to show the third point. Let F^s be a stationary point of the algorithm. It is sufficient to show that $\langle QF^s, Z - F^s\rangle \geq 0$ for all feasible vectors z (see [6]), where $\langle,\rangle$ denotes the standard inner product in $\mathbb{R}^n$. To show this we first note that $Q = D^T Q_2 D$, where

$$D = \begin{pmatrix} 1 & & & & & \\ -1 & 1 & & & & \\ & \ddots & \ddots & & & \\ & & & -1 & 1 & \\ & & & & -1 & 1 \end{pmatrix}$$

and Q_2 is the matrix according to QP3, i.e., to the direct problem. So we can deduce that $\langle QF^s, Z - F^s\rangle = (Z - F^s)^T QF^s = (Z - F^s)^T D^T Q_M DF^s = \langle Q_M f^s, z - f^s\rangle (f^s := DF^s, z := DZ)$. Now we only have to look at the "active points" because $(QF^s)_i$ is

zero everywhere else. Let Z be an arbitrary feasible vector and j be an index with $Z_j^s = L_j$ and $(Qz)_j \neq 0$, so $-\omega(QF^s) < 0$. With the feasibility of Z, it follows that $(QF^s)_j(Z_j - Z_j^s) = (QF^s)_j(Z_j - L_j) \geq 0$. With the same argument we can derive $(QF^s)_j(Z_j - Z_j^s) \geq 0$ if F^s touches the upper bound. It remains to show the inequality for the linearity intervals. Let $[t_\nu, t_{\nu+k}]$ be a linearity interval of F^s. Then obviously $[t_{\nu+1}, t_{\nu+k}]$ is a constancy interval for F^s. Furthermore it follows from the stationarity of F^s that a, b as defined in the algorithm are zero. This is equivalent to

$$\sum_{i=\nu}^{\nu+k}(QF^s) = 0, \qquad \sum_{i=\nu}^{\nu+k} t_i QF^s = \frac{1}{n}\sum_{i=\nu}^{\nu+k} iQF^s = 0 \quad (t_i = i/n),$$

which implies that

$$\sum_{i=1}^{l}(QX)_i = (Q_M x)_l, \qquad \sum_{i=\nu}^{l} i(QX)_i = \sum_{i=1}^{l}(Q_M x)_i$$

for arbitrary $X \in \mathbb{R}^n$ and $x = DX$. So our conditions are

$$(Q_M F^s)_\nu = 0, \qquad \sum_{i=\nu}^{\nu+k}(Q_M F^s)_i = 0 \Rightarrow \sum_{i=\nu+1}^{\nu+k}(Q_M F^s)_i = 0.$$

This case was proved by Löwendick [6]. □

Bibliography

1. William L. Briggs. *A Multigrid Tutorial.* SIAM, New York, 1994.
2. Paul H. Calamai and Jorge J. Moré. Projected gradient methods for linearly constrained problems. *Mathematical Programming*, 39:93–116, 1987.
3. P. L. Davies and A. Kovac. Modality, Runs, Strings and Multiresolution. *To appear in Annals of Statistics*, 2001.
4. D. L. Donoho and I.M. Johnstone. Ideal Spatial Adaption by Wavelet Shrinkage. *Biometrika*, 81:425–455, 1994.
5. W. Hackbush. *Multi-Grid Methods and their Applications.* Springer, Berlin, 1985.
6. M. Löwendick. On Smoothing under Bounds and Geometric Constraints. *Dissertation, Universität Essen*, 2000.
7. L. Metzner. Facettierte Nichtparametrische Regression. *Dissertation, Universität Essen*, 1997.
8. Jorge Nocedal and Stephen J. Wright. *Numerical Optimization.* Springer, Berlin, 1999.
9. Gerardo Toraldo and Jorge J. Moré. On the solution of large quadratic programming problems with bound constraints. *SIAM J. Optimization*, 1:93–113, 1991.