

# An Algorithm for Projections onto Convex Cones with Applications in Statistical Modelling

Mary C. Meyer

June 5, 2008

Problems with estimation under linear inequality constraints often arise in statistical modelling. In this paper we propose an algorithm to solve the quadratic programming problem of minimizing  $\psi(\boldsymbol{\theta}) = \boldsymbol{\theta}'Q\boldsymbol{\theta} - 2\mathbf{c}'\boldsymbol{\theta}$  with positive-definite  $Q$ , under the constraints  $A\boldsymbol{\theta} \geq 0$ . The problem is formulated as a projection onto a convex polyhedral cone. Applications such as shape-restricted regression and generalized regression, constrained parametric regression, and smoothed shape-restricted regression are discussed. Simulations show that the speed is favorable compared with the Mixed Primal-Dual Bases algorithm by Fraser and Massam (1989). The code is provided in R programming language for the general case and each of the specific applications listed in section 3.

*Keywords:* monotone regression, isotonic regression, convex regression, shape-restricted regression, constrained regression, quadratic programming, least-squares, inequality constraints.

# 1 Introduction and Statement of the Problem

Let  $Q$  be an  $n \times n$  positive-definite matrix, let  $\mathbf{c} \in \mathbb{R}^n$ , and let  $A$  be an  $m \times n$  full-row-rank constraint matrix. The objective function

$$\psi(\boldsymbol{\theta}) = \boldsymbol{\theta}'Q\boldsymbol{\theta} - 2\mathbf{c}'\boldsymbol{\theta} \tag{1}$$

is to be minimized over the set

$$\mathcal{C} = \{\boldsymbol{\theta} : A\boldsymbol{\theta} \geq 0\}. \tag{2}$$

(Note that when a vector is non-negative, all of its elements are non-negative.) The function  $\psi$  is strictly convex, and the constraint set  $\mathcal{C} = \{\boldsymbol{\theta} : A\boldsymbol{\theta} \geq 0\}$  is also convex, so that there is a unique minimum  $\hat{\boldsymbol{\theta}} \in \mathcal{C}$ . The interior point algorithm for minimizing a quadratic function over a convex set is a gradient-based algorithm, first proposed by Karmarkar (1984) for linear programming. From a feasible first guess, the algorithm moves along the gradient towards the boundary of the set, stopping at a point still in the interior. The set is mapped onto itself to bring current solution closer to the middle. The algorithm iterates until a tolerance is reached. For more details, see Fang and Puthenpura (1993), chapters 9 and 10.

The interior point algorithm is considered to convergence in “infinitely many” steps, because the true solution is approached asymptotically and never reached except within a user-defined tolerance. In contrast, the mixed primal-dual bases algorithm of Fraser and Massam (1989) is guaranteed to produce the solution in a finite number of steps. It is less general than the interior point method in that it requires the convex set to be in the form (2), which can be seen to be a convex polyhedral cone. The edges or generators of the cone and its polar cone are exploited.

The proposed new “hinge” algorithm is also a finite-step algorithm; it and the mixed primal-dual bases algorithm are outlined in the next section. Applications and simulations follow in Section 3, and some proofs are found in the appendix.

## 2 Algorithms for Cone Projection

Each row of the  $n \times m$  constraint matrix  $A$  defines a half-space in  $\mathbb{R}^n$  and the set  $\mathcal{C} = \{\boldsymbol{\theta} : A\boldsymbol{\theta} \geq 0\}$  is the intersection of these half-spaces. Thus,  $\mathcal{C}$  is a convex, polyhedral cone with vertex at the origin. The cone contains a linear space  $V$ , of dimension  $n - m$ ; this is the null space of  $A$ . Let  $\Omega = \mathcal{C} \cap V^\perp$ ;  $\Omega$  is again a polyhedral convex cone that does not contain a linear space of dimension one or greater, and sits in an  $m$ -dimensional subspace of  $\mathbb{R}^n$ . The edges or generators of  $\Omega$  are vectors  $\boldsymbol{\delta}^1, \dots, \boldsymbol{\delta}^m$  in  $\Omega$  such that

$$\Omega = \left\{ \boldsymbol{\theta} \in \mathbb{R}^n : \boldsymbol{\theta} = \sum_{i=1}^m b_i \boldsymbol{\delta}^i, \text{ where } b_i \geq 0, i = 1, \dots, m \right\},$$

and hence

$$\mathcal{C} = \left\{ \boldsymbol{\theta} \in \mathbb{R}^n : \boldsymbol{\theta} = \mathbf{v} + \sum_{i=1}^m b_i \boldsymbol{\delta}^i, \text{ where } b_i \geq 0, i = 1, \dots, m \text{ and } \mathbf{v} \in V \right\}. \quad (3)$$

**Proposition 1** *The edges of  $\Omega$  are the columns of the matrix  $\Delta = A'(AA')^{-1}$ .*

*Proof:* Note that the columns of  $\Delta$  are linearly independent and span the same space as the rows of  $A$ , hence are orthogonal to  $V$ . Suppose  $\boldsymbol{\theta} = \mathbf{v} + \Delta\mathbf{b}$  where  $\mathbf{b} \geq 0$  and  $\mathbf{v} \in V$ . Then  $A\boldsymbol{\theta} = A\mathbf{v} + A\Delta\mathbf{b} = \mathbf{b}$ , so  $\boldsymbol{\theta} \in \mathcal{C}$ . Alternatively, suppose  $A\boldsymbol{\theta} \geq 0$  and write  $\boldsymbol{\theta} = \mathbf{v} + \Delta\mathbf{b}$  where  $\mathbf{v}$  is the projection of  $\boldsymbol{\theta}$  on  $V$ . The term  $\Delta\mathbf{b}$  is the projection of  $\boldsymbol{\theta}$  on the linear space containing  $\Omega$ , so  $\mathbf{b} = (\Delta'\Delta)^{-1}\Delta'\boldsymbol{\theta} = A\boldsymbol{\theta} \geq 0$ .

◇

Necessary and sufficient conditions for  $\hat{\boldsymbol{\theta}}$  to minimize  $\|\mathbf{y} - \boldsymbol{\theta}\|^2$  are

$$\langle \mathbf{y} - \hat{\boldsymbol{\theta}}, \hat{\boldsymbol{\theta}} \rangle = 0 \text{ and } \langle \mathbf{y} - \hat{\boldsymbol{\theta}}, \boldsymbol{\theta} \rangle \leq 0 \text{ for all } \boldsymbol{\theta} \in \Omega; \quad (4)$$

see Robertson, Wright, and Dykstra (1988), p 15. For the next few results, we consider the special case that  $Q = I$ , the identity matrix, so that the solution  $\hat{\boldsymbol{\theta}}$  of (1) is the projection of

$\mathbf{y}$  onto  $\mathcal{C}$ , i.e., the minimizer of  $\|\mathbf{y} - \boldsymbol{\theta}\|^2$  over  $\boldsymbol{\theta} \in \mathcal{C}$ . Because  $V$  is orthogonal to  $\Omega$ ,  $\hat{\boldsymbol{\theta}}$  is the sum of the projections of  $\mathbf{y}$  onto  $V$  and  $\Omega$ . Then  $\hat{\boldsymbol{\theta}}$  can be written as  $\mathbf{v} + \Delta\mathbf{b}$ , where  $\mathbf{b} \geq 0$ .

**Proposition 2** *Let  $\hat{\boldsymbol{\theta}}$  be the unique minimizer of  $\|\mathbf{y} - \boldsymbol{\theta}\|^2$  over  $\boldsymbol{\theta} \in \mathcal{C}$ , and write  $\hat{\boldsymbol{\theta}} = \mathbf{v} + \Delta\mathbf{b}$  for  $\mathbf{b} \geq 0$ . Let  $J \subseteq \{1, \dots, m\}$  index the non-zero elements of  $\mathbf{b}$ ; that is,  $j \in J$  if  $b_j > 0$ . Then  $\hat{\boldsymbol{\theta}}$  is the projection of  $\mathbf{y}$  onto the linear space spanned by  $\boldsymbol{\delta}^j$ ,  $j \in J$ , and the basis vectors for  $V$ .*

*Proof:* We need only show that the projection of  $\mathbf{y}$  onto  $\Omega$  is the projection of  $\mathbf{y}$  onto the linear space spanned by  $\boldsymbol{\delta}^j$ ,  $j \in J$ . By (4), we must have  $\langle \mathbf{y} - \hat{\boldsymbol{\theta}}, \boldsymbol{\delta}^l \rangle \leq 0$  for  $l = 1, \dots, m$ . Suppose that for some  $l \in J$ , we have  $\langle \mathbf{y} - \hat{\boldsymbol{\theta}}, \boldsymbol{\delta}^l \rangle < 0$ . Then for  $\epsilon \in [0, b_l]$ ,  $\tilde{\boldsymbol{\theta}} = \hat{\boldsymbol{\theta}} - \epsilon\boldsymbol{\delta}^l \in \Omega$ , and

$$\begin{aligned} \|\mathbf{y} - \hat{\boldsymbol{\theta}}\|^2 &= \|\mathbf{y} - \tilde{\boldsymbol{\theta}}\|^2 + \|\tilde{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}\|^2 + 2\langle \mathbf{y} - \tilde{\boldsymbol{\theta}}, \tilde{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}} \rangle \\ &= \|\mathbf{y} - \tilde{\boldsymbol{\theta}}\|^2 + \|\tilde{\boldsymbol{\theta}} - \hat{\boldsymbol{\theta}}\|^2 + 2\langle \mathbf{y} - \hat{\boldsymbol{\theta}} + \epsilon\boldsymbol{\delta}^l, -\epsilon\boldsymbol{\delta}^l \rangle \\ &= \|\mathbf{y} - \tilde{\boldsymbol{\theta}}\|^2 - 2\epsilon\langle \mathbf{y} - \hat{\boldsymbol{\theta}}, \boldsymbol{\delta}^l \rangle - \epsilon^2 \|\boldsymbol{\delta}^l\|^2. \end{aligned}$$

The inner product is negative, so that the sum of the second two terms on the right is positive for some  $\epsilon \in [0, b_l]$ . This contradicts the premise that  $\hat{\boldsymbol{\theta}}$  minimizes the sum squares in  $\mathcal{C}$ . Therefore, we must have  $\langle \mathbf{y} - \hat{\boldsymbol{\theta}}, \boldsymbol{\delta}^l \rangle = 0$  for all  $l \in J$ , and hence  $\hat{\boldsymbol{\theta}}$  is the projection onto the linear space.

◇

This tells us that finite-step algorithms exist; for each of the  $2^m$  sets  $J$ , the projection onto the linear space spanned by  $\boldsymbol{\delta}^j$ ,  $j \in J$  can be computed, until the solution satisfying (4) is found. The mixed primal-dual bases algorithm of Fraser and Massam (1989) and the proposed hinge algorithm both choose an initial guess  $J$ , then add or subtract elements until the correct  $J$  is found. The claim in this paper is that the hinge algorithm is more straight-forward to

understand and implement, and simulations using the convex regression algorithm show that the number of steps in the chain of  $J$  guesses is about half.

### The Hinge Algorithm

The proposed algorithm finds  $\hat{\theta}$  by arriving at the appropriate set  $J$  through a series of guesses  $J_k$ . At a typical iteration, the current estimate  $\theta^k$  is the least-squares regression of  $\mathbf{y}$  on the space spanned by  $\delta^j$ , for  $j \in J_k$ . The  $\delta^j$ ,  $j \in J$ , are called “hinges” since in the convex regression problem for which the algorithm was initially devised, the points  $(t_j, \theta_j)$ ,  $j \in J$ , are the points at which the line segments in the piecewise linear fit change slope, and the bends are allowed to go only one way. The number of bends in the solution coincides with the size of  $J$ . The initial guess  $J_0$  can be any subset of  $\{1, \dots, m\}$ , including the empty set, which is recommended for convex regression. The hinge algorithm can be summarized in four steps. At the  $k$ th iteration,

1. Fit the data to the set  $\{\delta^j, j \in J_k\}$ , to get a least-squares estimate  $\theta^k$ .
2. Compute  $\langle \mathbf{y} - \theta^k, \delta^j \rangle$  for each  $j \notin J_k$ . If these are all nonpositive, then stop. If not, then add the vector  $\delta^j$  to the model for which this inner product is largest.
3. Get the least-squares fit with the new set of  $\delta$ -vectors.
4. Check to see if the regression function satisfies the constraints, i.e. if all coefficients are non-negative:
  - If yes, go to step 2.
  - If no, choose the hinge with the largest negative coefficient and remove it from the set. Go to step 3.

Intuitively, at each stage, the new hinge is added where it is “most needed,” and other hinges are removed if the new fit does not satisfy the constraints. Since the stopping criteria

are defined by (4), it is clear that if the algorithm ends, it gives the correct solution. The only thing that requires proof is that the algorithm does end, that is, it does not produce the same set of hinges twice, which would result in an infinite loop. The proofs are deferred to the appendix.

An efficient implementation of the hinge algorithm uses a sequence of QR decompositions of the design matrix of  $\delta$ -vectors. Let the columns of the matrix  $\Delta_k$  contain the vectors  $\delta^j$ , for  $j \in J_k$ . The  $k$ th iterated solution  $\theta$  is the projection of  $\mathbf{y}$  onto the column space of  $\Delta_k$ . At each stage, we can compute the decomposition  $\Delta_k = \mathbf{Q}_k \mathbf{R}_k$ , where the columns of  $\mathbf{Q}_k$  form an ortho-normal set of vectors. Because the  $k + 1$ st design matrix varies from the  $k$ th by one column, it is easy and computationally efficient to obtain  $\mathbf{Q}_{k+1}$  and  $\mathbf{R}_{k+1}$  from  $\mathbf{Q}_k$  and  $\mathbf{R}_k$ . At each stage,  $\theta^k = \mathbf{Q}_k \mathbf{Q}'_k \mathbf{y}$ , so that no matrix inversions are necessary.

### The Mixed Primal-Dual Bases Algorithm

We present a short outline; see Fraser and Massam (1989) for more details including a proof of convergence. Several new concepts must be defined. First, the polar cone  $\Omega^\circ$  is the analog of the perpendicular space in linear regression, as the projection of the data vector onto the polar cone is the residual of the projection onto the constraint cone. The polar cone is defined as all vectors making obtuse angles with all vectors in  $\mathcal{C}$ :

$$\Omega^\circ = \{\boldsymbol{\rho} : \langle \boldsymbol{\rho}, \boldsymbol{\theta} \rangle \leq 0, \text{ for all } \boldsymbol{\theta} \in \mathcal{C}\}.$$

Alternatively, the polar cone may be defined as all vectors whose projection onto the constraint cone is the origin. Let the vectors  $\boldsymbol{\gamma}^1, \dots, \boldsymbol{\gamma}^m$  be the rows of  $\mathbf{A}$ , multiplied by negative one. Then the constraints  $\mathbf{A}\boldsymbol{\theta} \geq 0$  may be written as

$$\langle \boldsymbol{\gamma}^i, \boldsymbol{\theta} \rangle \leq 0, \text{ for } i = 1, \dots, m.$$

The vectors  $\gamma^j$  are the edges of the polar cone; that is,

$$\Omega^o = \left\{ \sum_{i=1}^m a_i \gamma^i, a_1, \dots, a_m \geq 0 \right\}.$$

Each subset  $J \subseteq \{1, \dots, m\}$  defines a *sector*

$$\Omega_J = \left\{ \mathbf{y} : \mathbf{y} = \sum_{j \in J} b_j \delta^j + \sum_{j \notin J} b_j \gamma^j \text{ with } b_j > 0 \text{ for } j \in J, \text{ and } b_j \geq 0 \text{ for } j \notin J \right\}.$$

The  $2^m$  sectors partition  $V^\perp$ . Fraser and Massam (1989) call the  $\gamma^j$ 's *primal* vectors and the  $\delta^j$ 's *dual* vectors. The mixed primal-dual bases algorithm finds the correct set  $J$  by moving along a line segment connecting the point  $\mathbf{z}_0 = \sum_{j=1}^m \delta^j$  with  $\mathbf{z}$ , where  $\mathbf{z}$  is  $\mathbf{y}$  projected onto the set  $V^\perp$ . At the  $k$ th iteration, the point  $\mathbf{z}^k$  on the line segment is reached; this point is also on the boundary of  $\Omega_{J_k}$ . The next iteration finds  $\mathbf{z}^{k+1}$ , farther along the segment, on the boundary of  $\Omega_{J_{k+1}}$ . At the beginning of the iteration, both  $\mathbf{z}$  and  $\mathbf{z}^k$  are expressed in the basis defined by  $J_k$ :

$$\mathbf{z} = \sum_{j \in J_k} b_j \delta^j + \sum_{j \notin J_k} b_j \gamma^j,$$

and

$$\mathbf{z}^k = \sum_{j \in J_k} a_j \delta^j + \sum_{j \notin J_k} a_j \gamma^j,$$

where  $a_j > 0$  for  $j \in J_k$  and  $a_j \leq 0$  for  $j \notin J_k$ . If  $b_j > 0$  for  $j \in J_k$  and  $b_j \leq 0$  for  $j \notin J_k$ , then the solution is reached and the algorithm stops. Otherwise, find

$$\mathbf{z}^{k+1} = \mathbf{z}^k + \alpha_{k+1}(\mathbf{z} - \mathbf{z}^k),$$

where  $\alpha_{k+1} \in (0, 1)$  is as large as possible while the coefficients of  $\mathbf{z}^{k+1}$  (still in the basis defined by  $J_k$ ) are all positive or nonnegative as they are in  $J_k$  or not, respectively. The point  $\mathbf{z}^{k+1}$  is on the face of  $\Omega_{J_k}$  which divides  $\Omega_{J_k}$  and  $\Omega_{J_{k+1}}$ . The algorithm terminates at the face of

the sector containing  $\mathbf{z}$ . It clearly takes a finite number of iterations since there are a finite number of sectors. For details, including a proof that  $\alpha_k \in (0, 1)$  for all  $k$ , see Fraser and Massam (1989).

At first glance, it may seem that the mixed primal-dual bases algorithm is much more computationally intensive than the hinge algorithm, since at each iteration, the coefficients for both  $\mathbf{z}$  and  $\mathbf{z}^k$  in the basis defined by  $J_k$  must be calculated. However, the coefficients for  $\mathbf{z}^{k+1}$  are trivially calculated from those for  $\mathbf{z}^k$ , and it turns out that it is not necessary to solve a system of  $m$  equations to get the coefficients for  $\mathbf{z}$  at each iteration. We can find the coefficients for the  $\gamma^j$ ,  $j \notin J_k$  from the fit to only the  $\delta^j$ ,  $j \in J_k$ , since for  $j \notin J_k$ ,

$$b_j = \langle \mathbf{z} - \hat{\boldsymbol{\theta}}^k, \boldsymbol{\delta}^j \rangle.$$

Similarly, we can find the coefficients for the  $\delta^j$ ,  $j \in J_k$  from the fit to only the  $\gamma^j$ ,  $j \notin J_k$ . In fact, it is not necessary to explicitly define both the primal and the dual vectors in the implementation of the algorithm.

### Quadratic programming

The quadratic programming problem (1) may be solved using the cone projection. Let  $LL'$  be the Cholesky decomposition of  $Q$ ,  $\boldsymbol{\phi} = L'\boldsymbol{\theta}$ , and  $\tilde{\mathbf{c}} = L^{-1}\mathbf{c}$ , so that (1) becomes

$$\psi(\boldsymbol{\phi}) = \|\tilde{\mathbf{c}} - \boldsymbol{\phi}\|^2,$$

and the constraints become  $\tilde{A}\boldsymbol{\phi} \geq 0$ , where  $\tilde{A} = A(L')^{-1}$ . The edges of the transformed cone are computed using Proposition 1, and  $\hat{\boldsymbol{\phi}}$  is the projection in the transformed space. Then the reverse transformation gives  $\hat{\boldsymbol{\theta}} = (L')^{-1}\hat{\boldsymbol{\phi}}$ .

### 3 Applications in Statistics

#### Shape-restricted least-squares regression

Suppose we have the regression model

$$y_i = f(x_i) + \sigma\epsilon_i, \quad i = 1, \dots, n, \quad (5)$$

where  $f$  is known to have some shape restriction such as monotonicity or convexity. If we write  $\theta_i = f(x_i)$ , the vector version of the model is

$$\mathbf{y} = \boldsymbol{\theta} + \sigma\boldsymbol{\epsilon},$$

and the constraints may be written as (2). For monotone constraints,  $m = n - 1$  and the non-zero elements of  $A$  are  $A_{i,i} = -1$  and  $A_{i,i+1} = 1$ , for  $i = 1, \dots, n - 1$ . For convexity,  $m = n - 2$  and the nonzero elements of  $A$  are  $A_{i,i} = x_{i+2} - x_{i+1}$ ,  $A_{i,i+1} = x_i - x_{i-2}$ , and  $A_{i,i+2} = x_{i+1} - x_i$ , for  $i = 1, \dots, n - 2$ . Other shape restrictions such as increasing and concave, or isotonic regression using a partial or quasi-order can also be expressed using such a set of linear inequality constraints. Any of these constitutes a quadratic programming problem with  $Q = I$  and  $\mathbf{c} = \mathbf{y}$ .

To illustrate the cone projection ideas for convex regression, the edge vectors  $\boldsymbol{\delta}^j$  for a dataset of size  $n = 8$  are shown in Figure 1, in plot (a). It is easy to see that any convex vector in  $\mathbb{R}^8$  must be a linear combination of these edge vectors with non-negative coefficients, plus an unrestricted linear combination of  $\mathbf{1}$  and  $\mathbf{x}$ . A scatterplot of data generated from a quadratic regression function is shown in (b), and the least-squares convex fit superimposed. For this dataset,  $J = \{1, 5\}$ , as seen by the placement of the “hinges” in the fit.

For a larger example, consider a dataset of size  $n = 50$  simulated from  $f(x) = x^2$  with *iid* normal random errors, shown in Figure 2. Suppose this is a “real” dataset collected by a scientist who knows only that the relationship between the expected value of  $y$  and  $x$  is

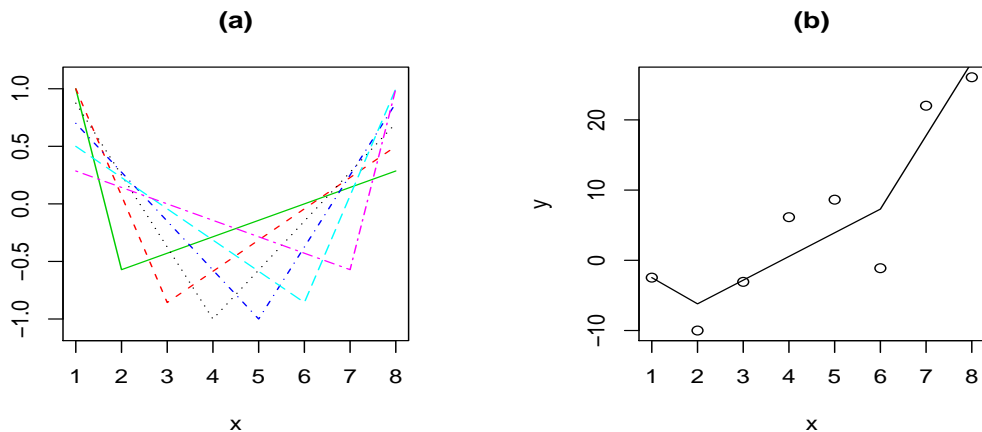


Figure 1: (a) The edge vectors  $\delta^1, \dots, \delta^6$  for convex regression with  $x_i = i$  and  $n = 8$ . Each is orthogonal to the space  $V$  spanned by the vectors  $\mathbf{1}$  and  $\mathbf{x}$ . (b) The least-squares convex fit to a dataset generated from a quadratic regression function. The fit uses the first and fifth edge vectors.

increasing. The least-squares fit using only monotonicity is shown in plot (a). If, instead, the scientist knows that the relationship must be convex, then the least-squares fit using this assumption is shown in (b). For increasing and convex assumptions, we obtain the fit in (c).

### Weighted shape-restricted regression

If  $\epsilon$  is mean zero multivariate normal with the positive-definite covariance matrix  $\Sigma$ , the maximum likelihood estimator for  $\theta$  minimizes the expression (1) where  $Q = \Sigma^{-1}$  and  $\mathbf{c} = \Sigma^{-1}\mathbf{y}$ . One simple use for the weighted model is in the case where the errors are *iid*, but  $x$ -values are not distinct. The construction of the constraint matrices and edge vectors requires that the  $t$ -values be distinct and ordered, but many data sets have multiple observations at a single value of the predictor. To compute the least-squares estimator, the  $y$ -values are averaged at each distinct  $x$ -value, then the weights applied to account for the smaller variance of an average of observations, compared to a single observation. Here, the  $\Sigma$  matrix is diagonal; the elements are the inverses of the numbers of observations at the corresponding distinct  $t$ -values.

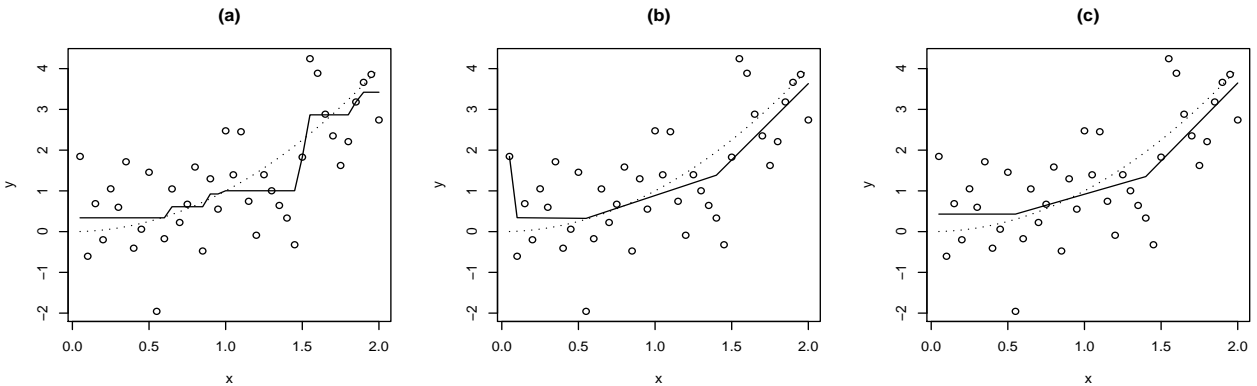


Figure 2: Least-squares fits to a scatterplot using shape-restrictions: (a) increasing, (b) convex, and (c) increasing and convex. The true function is shown as a dotted line.

### Shape-restricted regression splines

Unlike nonparametric function estimators that focus on smoothing, the shape-restricted regression estimator does not require user-defined choices such as bandwidth or smoothing parameter. However, some smoothing of the estimators might be desired. A straight-forward combination of smoothing and shape restrictions is found in regression splines (Meyer 2008). A set of knots must be chosen by the practitioner, but the shape restrictions provide robustness to knot choices. For degree- $d$  polynomial splines, a set of basis functions is defined given a vector of knot values, so that the shape restrictions are satisfied. These have the property that a degree- $d$  piecewise polynomial with the given knots satisfies the shape restrictions if and only if it is a linear combination of the basis functions with non-negative coefficients. Given a scatterplot of data, the basis vectors contain the values of the basis functions evaluated at the observed  $x$ -values. An example is shown in Figure 3, for piece-wise cubic polynomial splines constrained to be both increasing and concave. The basis functions are shown in plot (a), where the dots are the basis vectors. The basis vectors are used as the edge vectors in the hinge algorithm, where  $V$  is the one-dimensional space consisting of all multiples of the one-vector. The least-squares spline fit to the scatterplot shown in (b); this is the projection

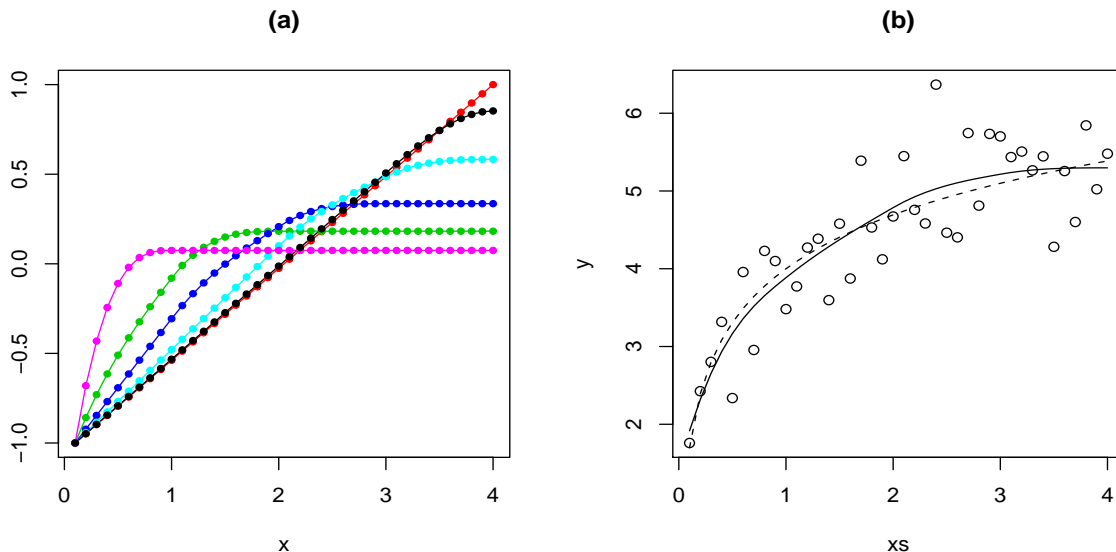


Figure 3: (a) Examples of spline basis functions (lines) and basis vectors (dots), for three equally spaced interior knots and  $n = 40$  equally spaced  $x$ -values, centered so that the basis vectors have mean zero. The fit to the scatterplot in (b) is the least-squares linear combination of the basis vectors with non-negative coefficients, plus an unconstrained multiple of the one-vector. The data were generated from  $f(x) = 4 + \log(x)$ , shown as the dotted curve.

of the data vector onto the cone defined by the edge vectors plus the projection of the data onto  $V$ .

### Constrained parametric regression

Suppose we have the usual linear regression model  $\mathbf{y} = X\boldsymbol{\beta} + \boldsymbol{\epsilon}$ , where the columns of the  $n \times k$  full-row-rank design matrix  $X$  contain values of predictor variables and  $\boldsymbol{\beta}$  is a  $k$ -dimensional parameter vector. We would like to obtain the least-squares estimator for  $\boldsymbol{\beta}$  under the constraints  $A\boldsymbol{\beta} \geq 0$ . For a simple, specific example, suppose our two predictor variables are time (continuous) and treatment (categorical). The response is expected to decrease linearly (or according to some other known functional form) with time, and there are three treatments

as well as a placebo. Specifically,

$$y_i = \alpha_0 + \beta_0 t_i + \beta_1 t_i d_{1i} + \beta_2 t_i d_{2i} + \beta_3 t_i d_{3i} + \epsilon_i,$$

where  $d_j$  is a dummy variable for treatment  $j$ ,  $\alpha_0$  is the common expected response at time zero,  $\beta_0$  is the slope for the placebo group, and  $\beta_0 + \beta_j$  is the slope for the  $j$ th treatment group. The covariance matrix for  $\epsilon$  is  $\sigma^2 \Sigma$  with  $\Sigma$  positive-definite. The researchers might want to constrain  $b_0 \leq 0$ , and  $b_j \leq b_0$ , for  $j = 1, 2, 3$ .

The problem is to minimize the weighted least-squares objective function

$$\beta' X' \Sigma^{-1} X \beta - 2 \mathbf{y}' \Sigma^{-1} X \beta$$

over  $A\beta \geq 0$ , which is in the form of (1) with  $\theta = \beta$ ,  $Q = X' \Sigma^{-1} X$ , and  $\mathbf{c} = X \Sigma^{-1} \mathbf{y}$ .

### Iteratively Re-weighted Least Squares

Shape-restricted function estimation is useful in generalized regression models. The standard logistic or Poisson models typically assume very specific functional relationships between the expected response and the predictor variable. Often these functional relationships are chosen for mathematical tractability and are not driven by theory. However, assumptions such as increasing probability curve or convex expected count are commonly valid.

The generalized linear regression models use iteratively re-weighted least squares projections to obtain the solution, and similarly, re-weighted projections onto cones can produce the maximum-likelihood solutions to the generalized shape-restricted regression problem. In general, the conditions for  $\hat{\theta}$  to uniquely maximize a function  $L(\theta)$  over a convex cone  $\mathcal{C}$  are (Robertson, Wright, and Dykstra (1988), p17)

$$\nabla L(\hat{\theta})' \hat{\theta} = 0, \text{ and} \tag{6}$$

$$\nabla L(\hat{\boldsymbol{\theta}})' \boldsymbol{\theta} \leq 0, \text{ for all } \boldsymbol{\theta} \in \mathcal{C}.$$

The symbol  $\nabla$  represents the gradient vector for the function. The latter condition can be written in terms of the edges of the constraint cone:

$$\nabla L(\hat{\boldsymbol{\theta}})' \boldsymbol{\delta}^j \leq 0, \text{ for } j = 1, \dots, m. \quad (7)$$

For the weighted least-squares projection, these conditions can be written as:

$$\sum_{i=1}^n w_i (y_i - \hat{\theta}_i) \hat{\theta}_i = 0,$$

and

$$\sum_{i=1}^n w_i (y_i - \hat{\theta}_i) \delta_i^j \leq 0, \text{ for } j = 1, \dots, m.$$

For many common models, the expressions (6) and (7) can be manipulated to mimic the least-squares conditions, and so suggest an iterative scheme. For example, consider the binomial model where the response variable takes on only two values such as success and failure, labeled one and zero. Suppose that the probability of success depends on the value of a predictor variable, say  $P(y = 1) = f(x)$ . Let the sorted, distinct values of the predictor variable be  $x_1, \dots, x_n$ , let  $k_1, \dots, k_n$  be the numbers of observations at the  $x$ -values, and let  $y_1, \dots, y_n$  be the numbers of successes. Then the log-likelihood function is

$$L(\boldsymbol{\theta}) = \sum_{i=1}^n \{y_i \theta_i + (k_i - y_i)[1 - \log(\theta_i)]\},$$

and

$$\nabla L(\hat{\boldsymbol{\theta}})' \boldsymbol{\theta} = \sum_{i=1}^n \left( \frac{y_i}{\hat{\theta}_i} - \frac{k_i - y_i}{1 - \hat{\theta}_i} \right) \theta_i.$$

the latter can be rearranged to look like

$$\sum_{i=1}^n \frac{k_i}{\hat{\theta}_i(1 - \hat{\theta}_i)} \left( \frac{y_i}{k_i} - \hat{\theta}_i \right) \theta_i.$$

This suggests an iterative scheme where an initial guess  $\theta^0$  is chosen, and iteration consists of defining “weights”  $k_i/[\hat{\theta}_i(1 - \hat{\theta}_i)]$  using the current iterate for  $\hat{\theta}$ , and “data”  $y_i/k_i$ . The algorithm ends when the conditions (6) and (7) are satisfied to within a small tolerance. The proof of convergence of this algorithm is given in Meyer (1999). For efficient coding, the initial guess  $J_0$  for the hinge algorithm should be taken to be the fitted  $J$  from the last iteration.

For example, consider the probability of being hospitalized after a motor vehicle accident, as a function of crash speed. It is reasonable to assume that this probability function is increasing with crash speed, but there might not be any reason to believe that it follows the logistic curve. An example using data from the National Highway Traffic Safety Administration is shown in Figure 4. The probability of hospitalization for drivers in motor vehicle crashes is to be estimated as a function of impact speed. The dataset is limited to drivers who are properly wearing their seatbelts, in “smaller” vehicles (sedans, compact, and sports cars) in years 1997-2002. Because the sample size is so large ( $n = 5549$  observations), the standard logistic regression model shows strong lack of fit. There are 95 observed impact speeds in kilometers per hour (kph); the points in Figure 4 represent the proportion injured at each observed speed. The speeds in the lower ranges have higher counts; there are only a few crashes at each impact speed over about 70 kph. For example, at 101 kph, there was only one crash, and (surprisingly) no serious injury sustained. The solid line represents the maximum likelihood estimate of the probability curve under the assumption that the probability of injury is increasing in impact speed. This curve was obtained through iteratively re-weighted projections onto the constraint cone associated with monotone regression.

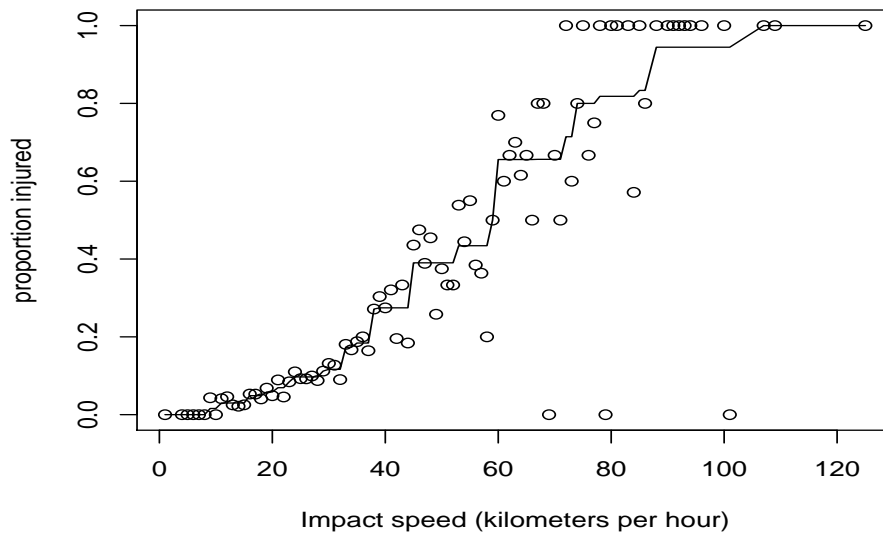


Figure 4: Probability of serious injury (hospitalization for more than one day) as a function of impact speed in motor vehicle crashes. Data are from the National Automotive Sampling System, Crashworthiness Data System, years 1997-2002. Observations are limited to front seat occupants aged 16 and older, in passenger cars in frontal collisions, who are properly wearing their seatbelts.

## Simulations

Simulations using convex regression show that the hinge algorithm requires fewer iterations to reach the solution, on average. The number of computations within each iteration are similar, so that the number of iterations of  $J_k$  can be compared to assess relative speeds. The number of iterations required for each algorithm depends on the size of the dataset and the size of the error. Large  $n$  and small errors both result in more hinges in the solution, and generally, the difference in iterations required to reach the solution is more dramatic in the case of many hinges. Table 1 compares the numbers of iterations required for the hinge algorithm with that for the mixed primal-dual bases algorithm. The data were generated as  $y_i = f(t_i) + \sigma\epsilon_i$ , for equally spaced  $t$ , where  $\epsilon_i$  are *i.i.d.* standard normal. Two choices each of sample size, error variance, and underlying regression function were used.

| <i>average # iterations, 1000 runs</i>      |       |      |
|---|-------|------|
| $f(t) = e^t, \sigma = 0.2$                  |       |      |
| n   | hinge | MPDB |
| 50  | 8.7   | 15.4 |
| 100   | 11.4  | 22.4 |
| $f(t) = e^t, \sigma = 0.05$                 |       |      |
| n   | hinge | MPDB |
| 50  | 11.5  | 21.6 |
| 100   | 16.0  | 31.4 |
| $f(t) = (t - \frac{1}{2})^2, \sigma = 0.1$  |       |      |
| n   | hinge | MPDB |
| 50  | 10.6  | 20.3 |
| 100   | 13.8  | 28.9 |
| $f(t) = (t - \frac{1}{2})^2, \sigma = 0.05$ |       |      |
| n   | hinge | MPDB |
| 50  | 12.6  | 24.4 |
| 100   | 17.1  | 32.8 |

Table 1: Average numbers of iterations required by the two algorithms for convex regression using simulated data.

## A Proofs

It is clear that if the hinge algorithm ends, it gives the correct solution. The algorithm ends because it does not choose the same set of hinges twice. The sum of squared errors (SSE)  $\| \mathbf{y} - \hat{\boldsymbol{\theta}} \|^2$  decreases for subsequent iterations with the same number of hinges, so that the algorithm cannot produce an infinite loop. First we show that the simplest type of loop does not occur.

**Proposition 3** *The algorithm does not remove the hinge that it just added.*

*Proof:* Suppose at the start of Step 2 in some iteration of the algorithm, the set of hinges is  $J_k$ , and at the end it is  $J_{k+1} = J_k \cup \{l\}$ , so that  $\boldsymbol{\delta}^l$  is the most recently added hinge. The coefficient of  $\boldsymbol{\delta}^l$  produced in Step 3 is

$$b_l = \frac{\langle \mathbf{y}, \tilde{\boldsymbol{\delta}}^l \rangle}{\| \tilde{\boldsymbol{\delta}}^l \|^2},$$

where  $\tilde{\boldsymbol{\delta}}^l$  is the residual from the regression of  $\boldsymbol{\delta}^l$  on the other regressors  $\{\boldsymbol{\delta}^j, j \in J_k\}$ . The numerator of the right hand side is equivalent to  $\langle \mathbf{y} - \hat{\boldsymbol{\theta}}^k, \boldsymbol{\delta}^l \rangle$ , because of orthogonality:

$$\langle \mathbf{y}, \tilde{\boldsymbol{\delta}}^l \rangle = \langle \mathbf{y} - \hat{\boldsymbol{\theta}}^k, \tilde{\boldsymbol{\delta}}^l \rangle = \langle \mathbf{y} - \hat{\boldsymbol{\theta}}^k, \boldsymbol{\delta}^l \rangle > 0.$$

The first equality is because  $\hat{\boldsymbol{\theta}}^k \perp \tilde{\boldsymbol{\delta}}^l$  and the second because  $\boldsymbol{\delta}^l - \tilde{\boldsymbol{\delta}}^l \perp \mathbf{y} - \hat{\boldsymbol{\theta}}^k$ .

◇

The idea for proving that the algorithm stops is to show that the sum of squares errors at a given iteration with  $n_h$  hinges is less than that of the last iteration with  $n_h$  hinges. Suppose that at the beginning of Step 2 for some iteration of the algorithm we have the solution:

$$\hat{\boldsymbol{\theta}}^B = \sum_{j \in J_B} b_j^B \boldsymbol{\delta}^j$$

and that this solution satisfies the constraints. Suppose that it is not optimal and the algorithm adds the vector  $\boldsymbol{\delta}^l$  to the set of regressors. The least-squares fit produced by Step 3 is:

$$\hat{\boldsymbol{\theta}}^M = \sum_{j \in \mathbf{J}_B} b_j^M \boldsymbol{\delta}^j + b_l^M \boldsymbol{\delta}^l.$$

Further suppose that this  $\hat{\boldsymbol{\theta}}^M$  does not satisfy the constraints, so that  $b_i^M$ , say, is negative. The algorithm will then remove  $\boldsymbol{\delta}^i$  from the set of regressors in Step 4 and go to Step 3 to refit the data. The next proposition shows that the new solution

$$\hat{\boldsymbol{\theta}}^N = \sum_{j \in \mathbf{J}_N} b_j^N \boldsymbol{\delta}^j,$$

where  $J_N = J_B \cup \{l\} - \{i\}$ , has  $SSE(\hat{\boldsymbol{\theta}}^N) < SSE(\hat{\boldsymbol{\theta}}^B)$ .

**Proposition 4** *If the algorithm replaces a hinge, then the SSE after is less than the SSE before.*

*Proof:* Let

$$\tilde{\boldsymbol{\delta}}^l = \boldsymbol{\delta}^l - \sum_{j \in \mathbf{J}_B} \alpha_j^l \boldsymbol{\delta}^j$$

where the second term is the projection of  $\boldsymbol{\delta}^l$  onto the space spanned by  $\{\boldsymbol{\delta}^j, j \in J_B\}$ . Then  $\tilde{\boldsymbol{\delta}}^l \perp \hat{\boldsymbol{\theta}}^B$  so we can write

$$\begin{aligned} \hat{\boldsymbol{\theta}}^M &= \hat{\boldsymbol{\theta}}^B + b_l^M \tilde{\boldsymbol{\delta}}^l \\ &= \sum_{j \in \mathbf{J}_B} (b_j^B - \alpha_j^l b_l^M) \boldsymbol{\delta}^j + b_l^M \boldsymbol{\delta}^l. \end{aligned}$$

We know that  $b_i^B > 0$  since  $\hat{\boldsymbol{\theta}}^B$  satisfies the constraints, and  $b_l^M > 0$ , by Proposition 3.

Further,

$$b_i^M = b_i^B - \alpha_i^l b_l^M < 0,$$

so that  $\alpha_i^l > 0$ . Let

$$\boldsymbol{\theta}(x) = \sum_{j \in \mathcal{J}_B} (b_j^B - \alpha_j^l x) \boldsymbol{\delta}^j + x \boldsymbol{\delta}^l.$$

Note that  $\boldsymbol{\theta}(0) = \hat{\boldsymbol{\theta}}^B$  and  $\boldsymbol{\theta}(b_l^M) = \hat{\boldsymbol{\theta}}^M$ . When  $x = b_i^B / \alpha_i^l$ , the coefficient of  $\boldsymbol{\delta}^i$  in  $\boldsymbol{\theta}(x)$  disappears. Further,  $0 < b_i^B / \alpha_i^l < b_l^M$ , since  $b_i^B - \alpha_i^l 0 > 0$  and  $b_i^B - \alpha_i^l b_l^M < 0$ . Since  $b_l^M$  minimizes  $\|\mathbf{y} - \boldsymbol{\theta}(x)\|^2$ , we have

$$\|\mathbf{y} - \boldsymbol{\theta}(0)\|^2 > \|\mathbf{y} - \boldsymbol{\theta}\left(\frac{b_i^B}{\alpha_i^l}\right)\|^2 > \|\mathbf{y} - \boldsymbol{\theta}(b_l^M)\|^2.$$

Further,

$$\|\mathbf{y} - \hat{\boldsymbol{\theta}}^N\|^2 < \|\mathbf{y} - \boldsymbol{\theta}\left(\frac{b_i^B}{\alpha_i^l}\right)\|^2$$

since  $\hat{\boldsymbol{\theta}}^N$  is the least-squares fit with the same regressors. So finally,

$$\|\mathbf{y} - \hat{\boldsymbol{\theta}}^N\|^2 < \|\mathbf{y} - \hat{\boldsymbol{\theta}}^B\|^2.$$

◇

## B Monotone Regression

There is the elegant closed-form solution for the monotone regression problem provided by Brunk (1955):

$$\theta_i = \min_{v \geq i} \max_{u \leq i} \frac{1}{v - u + 1} \sum_{j=u}^v y_j.$$

The pooled adjacent violators algorithm, known as PAVA, is an efficient and well-known method for finding this solution. See Robertson, Wright, and Dykstra (1988), p9 for details. The monotone regression problem is an interesting application of the hinge algorithm because during its implementation, no hinge indices are removed from the sets  $J_k$  at any iteration. Therefore, the number of iterations is the number of jumps in the monotone regression estimator.

The  $\delta$ -vectors can be written as:  $\delta_i^j = j - n$ , for  $i = 1, \dots, j$ , and  $\delta_i^j = j$ , for  $i = j + 1, \dots, n$ , for  $j = 1, \dots, m$ . Suppose at some iteration we have  $J_k = \{j_1, \dots, j_p\}$ , where the elements of  $J_k$  are ordered so that  $j_1 < \dots < j_p$ . Then it is easily seen that for  $1 \leq l \leq p$ ,

$$\bar{y} + b_{j_1} + \dots + b_{j_l} = \frac{1}{j_{l+1} - j_l} \sum_{i=j_l}^{j_{l+1}-1} y_i.$$

It can be shown that the first hinge index, say  $l$ , is chosen so that for any  $j_1 < l$  and  $j_2 > l$ ,

$$\frac{1}{l - j_1} \sum_{i=j_1}^{l-1} y_i \leq \frac{1}{j_2 - l} \sum_{i=l}^{j_2-1} y_i.$$

This means that the coefficient on the hinge  $\delta^l$  remains positive for any subsequent hinge, and a similar argument can be applied to that hinge, so that all coefficients remain positive throughout the implementation of the algorithm, and their indices are never removed from the sets  $J_k$ . From any  $J$ , the projection of  $\mathbf{y}$  onto the face  $\mathcal{F}_J$  is easily found by averaging the  $y$ -values between the  $j \in J$ .

## References

- [1] Brunk, (1955) Maximum likelihood estimates of monotone parameters. *Annals of Mathematical Statistics* **26(4)**, 607-616.
- [2] Fang, S.C. and Puthenpura, S. (1993) *Linear Optimization and Extensions. Theory and Algorithms*. Prentice Hall, Englewood Cliffs, New Jersey.
- [3] Fraser, D.A.S. and Massam, H., (1989). A mixed primal-dual bases algorithm for regression under inequality constraints. Application to convex regression. *Scand. J. Statist*, **16**, 65-74
- [4] Karmarkar, N. (1984) A new polynomial time algorithm for linear programming. *Combinatorica*, **4**, 373-395.
- [5] Meyer, M.C. (1999) A comparison of nonparametric shape-constrained bioassay estimators (1999), *Statistics and Probability Letters*, **42(3)** 267-274.
- [6] Meyer, M.C. (2008) Inference using shape-restricted regression splines. In press, *Annals of Applied Statistics*.
- [7] Robertson, T., Wright, F. T., and Dykstra, R. L. (1988) *Order Restricted Statistical Inference* John Wiley & Sons, New York