An Algorithm for Quadratic Programming with Applications in Statistics

Mary C. Meyer

October 27, 2009

Problems involving estimation and inference under linear inequality constraints arise often in statistical modelling. In this paper we propose an algorithm to solve the quadratic programming problem of minimizing $\psi(\boldsymbol{\theta}) = \boldsymbol{\theta}' Q \boldsymbol{\theta} - 2\boldsymbol{c}'\boldsymbol{\theta}$ for positive-definite Q, where $\boldsymbol{\theta}$ is constrained to be in a closed polyhedral convex cone $\mathcal{C} = \{\boldsymbol{\theta} : A\boldsymbol{\theta} \geq \boldsymbol{d}\}$, and the $m \times n$ matrix A is not necessarily full row-rank. Examples of applications include inequality-constrained parametric regression, and generalized shape-restricted regression using B-splines. Examples are given of optimization problems with more general ψ that can be solved with iterative quadratic programs. The three-step algorithm is intuitive and easy to code, and facilitates inference methods such as the $\bar{B}$ test. The method also provides an algorithm for isotonic regression that is substantially faster than the classic pooled adjacent violators algorithm. Code is provided in the `R` programming language.

Keywords: Constrained parameter estimation, shape-restricted regression, restricted least-squares, inequality constraints, primal-dual methods, $\bar{B}$ test.

1 Motivation and Background

Let $\boldsymbol{Q}$ be a positive definite $n \times n$ matrix, let $\boldsymbol{c}$ be a vector in $I\!R^n$, let $\boldsymbol{A}_0$ be an $m \times n$ irreducible matrix, and let $\boldsymbol{d}$ be a vector in the column space of $\boldsymbol{A}$. The term "irreducible" means "non-redundant" and will be defined formally in section 2. The quadratic programming problem

$$\text{find } \hat{\boldsymbol{\theta}} \text{ to minimize } \boldsymbol{\theta}' Q \boldsymbol{\theta} - 2\boldsymbol{c}'\boldsymbol{\theta} \text{ subject to } \boldsymbol{A}_0\boldsymbol{\theta} \geq \boldsymbol{d} \tag{1}$$

may be readily transformed into a "cone projection" problem by considering the Cholesky decomposition $\boldsymbol{U}'\boldsymbol{U} = \boldsymbol{Q}$ and finding $\boldsymbol{\theta}_0$ such that $\boldsymbol{A}_0\boldsymbol{\theta}_0 = \boldsymbol{d}$. Then define $\boldsymbol{\phi} = \boldsymbol{U}(\boldsymbol{\theta} - \boldsymbol{\theta}_0)$, $\boldsymbol{z} = (\boldsymbol{U}^{-1})'(\boldsymbol{c} - \boldsymbol{Q}\boldsymbol{\theta}_0)$, and $\boldsymbol{A} = \boldsymbol{A}_0\boldsymbol{U}^{-1}$ to get

$$\text{find } \hat{\boldsymbol{\phi}} \text{ to minimize } \parallel \boldsymbol{z} - \boldsymbol{\phi} \parallel^2 \text{ subject to } \boldsymbol{A}\boldsymbol{\phi} \geq \boldsymbol{0}. \tag{2}$$

The set

$$\mathcal{C} = \{\boldsymbol{\phi} \in I\!R^n : \boldsymbol{A}\boldsymbol{\phi} \geq \boldsymbol{0}\} \tag{3}$$

is easily seen to be a polyhedral convex cone in $\mathcal{R}^n$, as each row of $\boldsymbol{A}$ defines a half-space, and $\mathcal{C}$ is the intersection of these half-spaces. Because the convex objective function is to be minimized over a convex set, a unique solution $\hat{\boldsymbol{\phi}}$ exists.

An example of an application in statistics is the least-squares regression problem with linear inequality constraints on the coefficients. Let $\boldsymbol{X}$ be an $n \times k$ fixed design matrix and $\boldsymbol{\beta}$ be a k-dimensional parameter vector, and consider the model

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\epsilon} \tag{4}$$

where $E(\boldsymbol{\epsilon}) = \boldsymbol{0}$ and $\text{cov}(\boldsymbol{\epsilon}) = \sigma^2\boldsymbol{I}$. Suppose it is reasonable to impose constraints on the parameter vector in the form $\boldsymbol{A}_0\boldsymbol{\beta} \geq \boldsymbol{d}$, where $\boldsymbol{A}_0$ is an $m \times k$ irreducible matrix. The problem of minimizing $\parallel \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \parallel^2$ subject to the constraints is a quadratic programming problem with

$\boldsymbol{Q} = \boldsymbol{X}'\boldsymbol{X}$, and $\boldsymbol{c} = \boldsymbol{X}'\boldsymbol{y}$. The constrained least-squares solution provides smaller squared error loss compared with the unconstrained fit if the true function indeed satisfies the constraints, with equality only if the two estimators coincide (a short proof is provided in section 4). Tests of $H_0 : \boldsymbol{A}_0\boldsymbol{\beta} = \boldsymbol{d}$ versus $H_a : \boldsymbol{A}_0\boldsymbol{\beta} > \boldsymbol{d}$ use the $\bar{B}$ statistic (discussed in section 4), and have higher power than the corresponding F-test with $H_a : \boldsymbol{A}_0\boldsymbol{\beta} \neq \boldsymbol{d}$.

Methods for minimizing $\| \boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta} \|^2$ subject to various linear inequality and equality constraints were proposed by Judge and Takayama (1966); they used a simplex algorithm to obtain the optimal $\boldsymbol{\beta}$. Liew (1976) considered constraints $\boldsymbol{A}\boldsymbol{\beta} \geq \boldsymbol{d}$, and used the Dantzig-Cottle "principal pivoting" algorithm and provided an approximate variance-covariance matrix for $\hat{\boldsymbol{\beta}}$. The $\bar{\chi}^2$ statistic was defined for known model variance by Kudô (1963) for multi-variate analogues of one-sided tests, in particular for order-restricted inference concerning the means of several populations. The $\bar{B}$ statistic is a straight-forward extension to the case of unknown model variance. Gouriéroux, Holly, and Monfort (1982) investigated the properties of tests of $H_0 : \boldsymbol{A}\boldsymbol{\beta} = \boldsymbol{c}$ versus $H_a : \boldsymbol{A}\boldsymbol{\beta} > \boldsymbol{c}$, defining a Kuhn-Tucker test and deriving the asymptotic distribution when $\boldsymbol{A}$ is full row-rank. They give examples using one and two constraints. Raubertas, Lee, and Nordheim (1986) further investigated tests of hypothesis defined by linear inequality constraints. Wolak (1989) investigated properties of tests based on the $\bar{\chi}^2$ statistic for mixtures of linear inequality and equality constraints on the parameters. Hawkins (1994) considered the problem of fitting monotonic polynomials to data using a primal-dual algorithm to adjust the coefficients iteratively.

Tests involving ordering constraints for categorical data were considered by Robertson and Wright (1981). Dardanoni and Forcina (1998) considered maximum likelihood estimation of stochastically ordered distributions of discrete random variables. They formulate the problem as an iteratively re-weighted cone projection, and recommend the algorithm of Dykstra (1983). Bartolucci, Forcina, and Dardanoni (2001) estimate discrete bivariate distributions under positive quadrant dependence, also using an iteratively re-weighted cone projection scheme. The special case of stochastic ordering involving multivariate totally positive binary random variables was considered by Bartolucci and Forcina (2000). In this case there are more constraints than dimensions, and the authors use iteratively re-weighted quadratic programs. Hall and Præstgaard (2001) test for homogeneity in the mixed model under the constraint that a covariance matrix must be non-negative definite. Molenberghs and Verbeke (2007) also give a number of examples of estimation and testing in constrained parameter spaces. A thorough treatment of the theory and history of restricted parameter space estimation is in van Eeden (2006), and chapter 21 of Gourieroux and Monfort (1995) provides an exposition of estimation and testing under linear inequality constraints.

Another traditional use for constrained estimation and inference is for nonparametric regression with shape restrictions. Suppose interest is in fitting a scatterplot generated from

$$y_i = f(x_i) + \epsilon_i, \;\; i = 1, \ldots, n, \tag{5}$$

where a parametric form is unknown for f, but it may be assumed that f is non-decreasing. Defining $\theta_i = f(x_i)$, the problem is to find $\boldsymbol{\theta} \in I\!R^n$ to minimize $\| \boldsymbol{y} - \boldsymbol{\theta} \|^2$ given $\boldsymbol{A\theta} \geq \mathbf{0}$, where the non-zero elements of the $(n-1) \times n$ matrix $\boldsymbol{A}$ are $A_{i,i} = -1$ and $A_{i,i+1} = 1$, $i = 1, \ldots, n-1$. This model was first considered by Brunk (1955) who proposed the pooled adjacent violators algorithm (PAVA) to obtain the solution. The $\bar{B}$ test for constant versus monotone regression function is described in Silvapulle and Sen (2005), chapter 3. For estimation assuming f is convex, the non-zero elements of the $(n-2) \times n$ constraint matrix are $A_{i,i} = t_{i+2} - t_{i+1}$, $A_{i,i+1} = t_i - t_{i+2}$, and $A_{i,i+2} = t_{i+1} - t_i$. Fraser and Massam (1990) proposed the mixed primal-dual bases algorithm to solve the concave regression problem, and the $\bar{B}$ test for linear versus convex regression function was derived in Meyer (2003).

This paper introduces a simple, three-step algorithm for cone projection that is fast and intuitive, and can be applied to any of the above quadratic programming situations. When applied to monotone regression, it is considerably faster than PAVA. The code, shown in the

R programming language in Appendix A, is only a couple dozen lines. The speed is important for applications in iteratively re-weighted cone projection algorithms such as for generalized regression, and in estimating the mixing distribution used for the $\bar{B}$ test. The rest of the paper is organized as follows. Some necessary theory about cones and projections onto cones is outlined in the next section; the propositions can be skipped by readers interested only in the performance and applications of the algorithm. The theory is used in the convergence results (Appendix B) and in the outline of the $\bar{B}$ test statistic (section 4); more details and proofs of the propositions can be found in Silvapulle and Sen (2005), chapter 3, or Meyer (1999). The proposed cone projection ("hinge") algorithm is described in the section 3 and compared to other algorithms. Examples of useful applications in statistics follow in section 4.

2 Properties of Convex Polyhedral Cones

The convex polyhedral cone (3) is considered for an $m \times n$ irreducible constraint matrix $\boldsymbol{A}$, where "irreducible," means that no row of $\boldsymbol{A}$ is a positive linear combination of other rows, and there is not a positive linear combination of rows of $\boldsymbol{A}$ that equals the zero vector. Note that if $\boldsymbol{A}$ is full row-rank, then it is irreducible. If a row is a positive linear combination of other rows, it can be removed without affecting the problem, and if the origin can be written as a positive linear combination of rows, then there is an implicit equality constraint in the matrix $\boldsymbol{A}$, which can be dealt with separately. Silvapulle and Sen (2005) use the term "tight" to describe this non-redundancy of $\boldsymbol{A}$.

Let m_1 be the dimension of the space spanned by the rows of A. If m_1 is less than n, then the cone contains a linear space V, of dimension $n - m_1$; this is the null space of A. Let $\Omega = \mathcal{C} \cap V^{\perp}$; then Ω is a polyhedral convex cone that does not contain a linear space of dimension one or greater, and sits in an m_1-dimensional subspace of $I\!R^n$. The "edges" or "generators" of Ω are vectors $\boldsymbol{\delta}^1, \ldots, \boldsymbol{\delta}^{m_2}$ in Ω such that

$$\Omega = \left\{ \boldsymbol{\phi} \in I\!R^n : \boldsymbol{\phi} = \sum_{i=1}^{m_2} b_j \boldsymbol{\delta}^j, \text{ where } b_j \geq 0, \; j = 1, \ldots, m_2 \right\},$$

and hence

$$\mathcal{C} = \left\{ \boldsymbol{\phi} \in I\!R^n : \boldsymbol{\phi} = \boldsymbol{v} + \sum_{i=1}^{m_2} b_j \boldsymbol{\delta}^j, \text{ where } b_j \geq 0, \; j = 1, \ldots, m_2 \text{ and } \boldsymbol{v} \in V \right\}. \tag{6}$$

If $m_1 = m$, then $m_2 = m$ and the edges of Ω are the columns of the matrix $\Delta = A'(AA')^{-1}$. For the case of more constraints than dimensions ($m > m_1$), the edges of the cone are obtained as follows (see Meyer 1999 for proof). Define $\boldsymbol{\gamma}^1, \ldots, \boldsymbol{\gamma}^m$ to be the rows of $\boldsymbol{A}$. Suppose $J \subseteq \{1, \ldots, m\}$ and let $S = \mathcal{L}(\{\boldsymbol{\gamma}^j, j \in J\})$, where $\mathcal{L}$ denotes "the space spanned by." If $\dim(S) = m_1 - 1$, then $S^\perp \cap V^\perp$ is a line through the origin containing the vectors $\boldsymbol{\delta}$ and $-\boldsymbol{\delta}$, say, where $\boldsymbol{\delta} \perp \boldsymbol{\gamma}^j$ for all $j \in J$. If $\langle \boldsymbol{\delta}, \boldsymbol{\gamma}^i \rangle \geq 0$ for all $i \notin J$, then $\boldsymbol{\delta}$ is an edge of $\boldsymbol{\Omega}$. Conversely, all edges are of this form. In the case of more constraints than dimensions, the number m_2 of edges may be considerably larger than m.

Because V is orthogonal to Ω, $\hat{\boldsymbol{\phi}}$ is the sum of the projections of $\boldsymbol{y}$ onto V and Ω, so that $\hat{\boldsymbol{\phi}}$ can be written as $\boldsymbol{v} + \Delta \boldsymbol{b}$, where $\boldsymbol{v} \in V$, the columns of $\boldsymbol{\Delta}$ are $\boldsymbol{\delta}_1, \ldots, \boldsymbol{\delta}_m$, and $\boldsymbol{b} \geq 0$. Note that this representation is not necessarily unique if there are more constraints than dimensions, but $\hat{\boldsymbol{\phi}}$ is unique. Necessary and sufficient conditions for $\hat{\boldsymbol{\phi}} \in \mathcal{C}$ to minimize $\| \boldsymbol{y} - \boldsymbol{\phi} \|^2$ are $\langle \boldsymbol{y} - \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\phi}} \rangle = 0$ and $\langle \boldsymbol{y} - \hat{\boldsymbol{\phi}}, \boldsymbol{\phi} \rangle \leq 0$, for all $\boldsymbol{\phi} \in \mathcal{C}$; because the $\boldsymbol{\delta}$ vectors are generators of the cone, these conditions may be written

$$\langle \boldsymbol{y} - \hat{\boldsymbol{\phi}}, \hat{\boldsymbol{\phi}} \rangle = 0; \; \langle \boldsymbol{y} - \hat{\boldsymbol{\phi}}, \boldsymbol{v} \rangle \leq 0 \text{ for all } \boldsymbol{v} \in V; \text{ and } \langle \boldsymbol{y} - \hat{\boldsymbol{\phi}}, \boldsymbol{\delta}^j \rangle \leq 0 \text{ for } j = 1, \ldots, m_2; \tag{7}$$

Proposition 1 *Let $\hat{\boldsymbol{\phi}}$ be the unique minimizer of $\| \boldsymbol{y} - \boldsymbol{\phi} \|^2$ over $\boldsymbol{\phi} \in \mathcal{C}$, and write $\hat{\boldsymbol{\phi}} = \boldsymbol{v} + \Delta \boldsymbol{b}$ for $\boldsymbol{b} \geq 0$ and $\boldsymbol{v} \in V$. Let $J \subseteq \{1, \ldots, m\}$ index the non-zero elements of $\boldsymbol{b}$; that is, $j \in J$ if*

$b_j > 0$. *Then $\hat{\boldsymbol{\phi}}$ is the projection of $\boldsymbol{y}$ onto the linear space spanned by $\boldsymbol{\delta}^j$, $j \in J$, and the basis vectors for V.*

Hence, the projection of a vector onto a polyhedral convex cone is the ordinary least-squares projection onto the linear space spanned by a basis for V and a subset of cone edges indexed by a subset J of $\{1, \ldots, m\}$. The projection lands on a *face* of the cone; the faces can be defined as

$$\mathcal{F}_J = \left\{ \boldsymbol{\phi} \in I\!\!R^n : \boldsymbol{\phi} = \boldsymbol{v} + \sum_{j \in J} b_j \boldsymbol{\delta}_j, \;\; b_j > 0, \;\; \boldsymbol{v} \in V \right\}.$$

Note that $\mathcal{F}_\emptyset = V$ (where $\emptyset$ denotes the empty set), and the interior of the constraint cone is the face where $J = \{1, \ldots, m\}$. Finally, we define *sectors* $\mathcal{C}_J$, containing all $\boldsymbol{y} \in I\!\!R^n$ such that the projection of $\boldsymbol{y}$ onto $\mathcal{C}$ falls on $\mathcal{F}_J$. The sectors are themselves polyhedral convex cones, and if $m = m_1$, the sectors partition $I\!\!R^n$. Otherwise, for irreducible A, any sector overlap has measure zero in $I\!\!R^n$.

The set $\mathcal{C}_\emptyset \cap V^\perp$ is also called the *polar cone.* The edges of the polar cone are simply the rows of $-\boldsymbol{A}$. The next proposition implies that the projection onto the constraint cone may alternatively be found via projection onto the polar cone.

Proposition 2 *Let $\hat{\boldsymbol{\rho}}$ be the projection of $\boldsymbol{y}$ onto the polar cone, and let $\hat{\boldsymbol{\phi}}$ be the projection of $\boldsymbol{y}$ onto the constraint cone. Then $\hat{\boldsymbol{\phi}} + \hat{\boldsymbol{\rho}} = \boldsymbol{y}$.*

3 Algorithms for Cone Projection

An early algorithm for cone projection was proposed by Dykstra (1983). Using that $\mathcal{C}$ is the intersection of half-spaces, it cyclically projects a residual vector onto the half-spaces, updating the residual vector at each step. 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. Fraser and Massam (1989) developed the mixed primal-dual bases algorithm for cone projections and applied it to concave nonparametric regression.

The interior point algorithm and the cyclical projections algorithm of Dykstra (1983) are considered to convergence in "infinitely many" steps, because the true solution is approached asymptotically and reached within a user-defined tolerance. In contrast, other algorithms that exploit the edges of the cone are guaranteed to produce the solution in a finite number of steps. These include the non-negative least squares (NNLS) algorithm of Lawson and Hanson (1995), the mixed primal-dual bases (MPDB) algorithm of Fraser and Massam (1989), and the "hinge" algorithm, proposed here.

The Hinge Algorithm

Proposition 1 tells us that the minimizer of (2) subject to $\boldsymbol{A\phi} \geq \mathbf{0}$ with irreducible $m \times n$ $\boldsymbol{A}$ can be solved by finding $J \subseteq \{1, \ldots, m\}$ where $\hat{\boldsymbol{\phi}}$ lands on $\mathcal{F}_J$, i.e., where $\boldsymbol{y} \in \mathcal{C}_J$. The hinge algorithm arrives at the appropriate set J through a series of guesses J_k. At a typical iteration, the current estimate $\boldsymbol{\phi}^k$ is the least-squares regression of $\boldsymbol{z}$ on the space spanned by $\boldsymbol{\delta}^j$, for $j \in J_k$. (The $\boldsymbol{\delta}^j$, $j \in J$, were originally called "hinges" since in the convex regression problem for which the algorithm was initially devised, the points (t_j, ϕ_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 initial guess J_0 can be any subset of $\{1, \ldots, m\}$ for which the corresponding $\boldsymbol{\delta}^j$, $j \in J$, form a linearly independent set. The hinge algorithm can be summarized in three steps. At the kth iteration,

1. Project $\boldsymbol{z}$ onto the linear space spanned by $\{\boldsymbol{\delta}^j, j \in J_k\}$, to get $\boldsymbol{\phi}^k = \sum_{j \in J_k} b_j^{(k)} \boldsymbol{\delta}_j$.

2. Check to see if $\boldsymbol{\phi}^k$ satisfies the constraints, i.e. if all $b_j^{(k)}$ are non-negative:

   - If yes, go to step 3.

- If no, choose j for which $b_j^{(k)}$ is smallest, and remove it from the set; go to step 1.

3. Compute $\langle \boldsymbol{y} - \boldsymbol{\phi}^k, \boldsymbol{\delta}^j \rangle$ for each $j \notin J_k$. If these are all nonpositive, then stop. If not, choose j for which this inner product is largest, add it to the set, and go to step 1.

Intuitively, at each stage, the new edge is added where it is "most needed," and other edges are removed if the new fit does not satisfy the constraints. Although the $\boldsymbol{\delta}^j$, $j = 1 \ldots, m_2$ might not form a linearly independent set, at each step the $\boldsymbol{\delta}^j$, $j \in J_k$ are linearly independent. For, suppose that at step 2, we have that J_k defines a linearly independent set of $\boldsymbol{\delta}^j$ vectors. For all vectors $\boldsymbol{\delta}^j$ such that the indices $J_k \cup \{j\}$ do not define a linearly independent set, we have $\langle \boldsymbol{y} - \boldsymbol{\phi}^k, \boldsymbol{\delta}^j \rangle = 0$, so these j are not added. Note that by Proposition 3, we can alternatively project onto the polar cone using the rows of $-\boldsymbol{A}$ as the edges in the algorithm. This is useful for problems with more constraints than dimensions, as the number of constraint cone edges can be large, and finding them may be computationally intensive.

Since the stopping criteria are defined by (7), 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 edges twice, which would result in an infinite loop. The proofs are deferred to the appendix.

The Nonnegative Least Squares Algorithm

The problem is stated in Lawson and Hanson (1995), chapter 23, as minimizing $\| \boldsymbol{y} - \Delta \boldsymbol{b} \|^2$ subject to $\boldsymbol{b} \geq \boldsymbol{0}$. The 12-step algorithm consists of two loops, one to add constraints and one to remove constraints from the active set. The authors argue that each loop must terminate in a finite number of steps, and hence the algorithm converges, but they do not show that the algorithm does not go back and forth between loops indefinitely.

The Mixed Primal-Dual Bases Algorithm

The mixed primal-dual bases (MPDB) algorithm of Fraser and Massam arrives at J in an intuitive way. The algorithm starts by choosing a vector $\boldsymbol{z}_0$ in a sector indexed by J_0, and moves toward the data vector $\boldsymbol{z}$ along the line connecting $\boldsymbol{z}_0$ and $\boldsymbol{z}$. Each time a sector boundary is crossed, the set J_k indexing the new sector is updated, until the sector containing $\boldsymbol{z}$ is reached. Because there are only a finite number of sectors, the algorithm is guaranteed to converge. The MPDB algorithm was extended to the case of more constraints than dimensions in Meyer (1999).

Comparison of Algorithms

The NNLS, MPDB, and hinge algorithms discover the set J indexing the sector containing $\boldsymbol{y}$. Each step involves an ordinary least-squares projection, so the fastest algorithm has fewest sets J_k in the chain. Once the NNLS problem is converted into a cone projection, this and the hinge algorithms require approximately the same computing time to converge if the starting J_0 is the null set, and in fact will build the set J in the identical sequence if the edge vectors are scaled to have unit lengths. Both tend to be faster than the MPDB, in our simulations using convex regression. The data were generated as $y_i = f(t_i) + \sigma\epsilon_i$, for equally spaced t, where ϵ_i are *iid* standard normal, for two choices each of sample size, error variance, and underlying regression function. For each combination, 1000 data sets were generated, and the algorithms performed on an Apple laptop, with a 2.4 GHz Intel processor. 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 edges in the solution, and generally, the difference in iterations required to reach the solution is more dramatic in these cases. Table 1 compares the total user time required for the hinge algorithm with that for the mixed primal-dual bases algorithm.

The hinge algorithm is more intuitive than the NNLS, and the `R` code listed in Appendix A is quite short. The option of starting with an arbitrary J is appealing for many applications in constrained smoothing. For applications such as standard monotone or convex regression where the final J is expected to be relatively small compared to m, the empty set is recommended for

	$f(t)=e^t$, $\sigma=0.2$		$f(t)=e^t$, $\sigma=0.05$	
n	hinge	MPDB	hinge	MPDB
50	2.82	4.98	4.65	9.15
100	4.60	8.65	7.82	15.21
	$f(t)=(t-\frac{1}{2})^2$, $\sigma=0.1$		$f(t)=(t-\frac{1}{2})^2$, $\sigma=0.05$	
n	hinge	MPDB	hinge	MPDB
50	3.87	7.41	6.21	13.31
100	5.05	9.77	8.03	15.20

Table 1: User time on MacBook Pro required by the two algorithms for convex regression using 1000 simulated data sets per combination of regression function and model variance.

J_0. If the set J is expected to be relatively large, such as for regression splines or constrained parametric function estimation, we can start with $J=\{1,\ldots,m\}$. For applications with iteratively re-weighted cone projections, the speed is considerably improved if the starting J was the final J for the last set of weights.

4 Applications in Statistics

Constrained Least-squares Regression

Consider the model (4) where it is known that $\boldsymbol{A\beta}\geq\mathbf{0}$ for irreducible $\boldsymbol{A}$. Let $\hat{\boldsymbol{\beta}}$ be the constrained least-squares estimator for $\boldsymbol{\beta}$, and let $\tilde{\boldsymbol{\beta}}$ be the unconstrained estimator $(\boldsymbol{X}'\boldsymbol{X})^{-1}\boldsymbol{X}'\boldsymbol{y}$. The Kuhn-Tuker conditions for the cone projection lead to $(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})'\boldsymbol{X}\hat{\boldsymbol{\beta}}=0$ and $(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})'\boldsymbol{X}\boldsymbol{\beta}\leq 0$ for all $\boldsymbol{\beta}$ such that $\boldsymbol{A\beta}\geq\mathbf{0}$. Now

$$\parallel \boldsymbol{X}\tilde{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}\parallel^2=\parallel \boldsymbol{X}\tilde{\boldsymbol{\beta}}-\boldsymbol{X}\hat{\boldsymbol{\beta}}\parallel^2+\parallel \boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}\parallel^2+2(\boldsymbol{X}\tilde{\boldsymbol{\beta}}-\boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}).$$

The last term is

$$(\boldsymbol{X}\tilde{\boldsymbol{\beta}}-\boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta})=(\boldsymbol{y}-\boldsymbol{X}\hat{\boldsymbol{\beta}})'(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta})-(\boldsymbol{y}-\boldsymbol{X}\tilde{\boldsymbol{\beta}})'(\boldsymbol{X}\hat{\boldsymbol{\beta}}-\boldsymbol{X}\boldsymbol{\beta}),$$

where the second term on the right is zero by principles of ordinary least-squares regression, and the first is positive by (7). Hence, if the true $\boldsymbol{\beta}$ satisfies the constraints, the constrained estimate of $E(\boldsymbol{y})$ is closer to the truth than the unconstrained estimate, with equality only if the two estimates coincide.

A well-known example is the simple linear regression model $y_i = \beta_0 + \beta_1 x_i + \epsilon_i$, where the slope β_1 must be non-negative. The constrained least-squares estimate is easy to obtain: it is the ordinary least-squares estimate unless that does not provide a positive slope, in which case $\hat{\beta}_1 = 0$ and $\hat{\beta}_0 = \bar{y}$. Now suppose the researcher wishes to allow some curvature in the regression function, so that the quadratic model $y_i = \beta_0 + \beta_1 x_i + \beta_2 x_i^2 + \epsilon_i$ is considered, retaining the assumption that the expected value of y is non-decreasing in x. The constraint is $\beta_1 + 2\beta_2 x \geq 0$ over the range of the data, so that if $x \in [0, 1]$, these can be written as $\boldsymbol{A}\boldsymbol{\beta} \geq \mathbf{0}$ with

$$\boldsymbol{A} = \begin{pmatrix} 0 & 1 & 0 \\ 0 & 1 & 2 \end{pmatrix},$$

and $\hat{\boldsymbol{\beta}}$ is obtained using the hinge algorithm. To illustrate, the scatterplot of $n = 50$ points shown in Figure 1 was generated using equally spaced x_i, the regression function $f(x_i) = 1 - (1 - x_i)^2$, and independent standard normal errors. Least-squares quadratic fits are shown, where the dashed curve represents the unconstrained fit and the solid curve is constrained to be monotone. The dotted curve is the true regression function.

Another important example is the "warped plane" model $y_i = \beta_0 + \beta_1 x_{1i} + \beta_2 x_{2i} + \beta_3 x_{1i} x_{2i} + \epsilon_i$, where $E(y)$ is constrained to be increasing in both variables. The constraints are $\beta_1 + \beta_3 x_{2i} \geq 0$, for all x_{2i} and $\beta_2 + \beta_3 x_{1i} \geq 0$, for all x_{1i}. If both predictors are confined to the unit interval, this provides four constraints on three parameters. Constrained and unconstrained fits are shown in Figure 2 for data simulated from the surface $f(x_1, x_2) = x_1 x_2$, unit error variance, and $n = 100$, with the predictor values forming a grid in the unit intervals.

Hypothesis testing

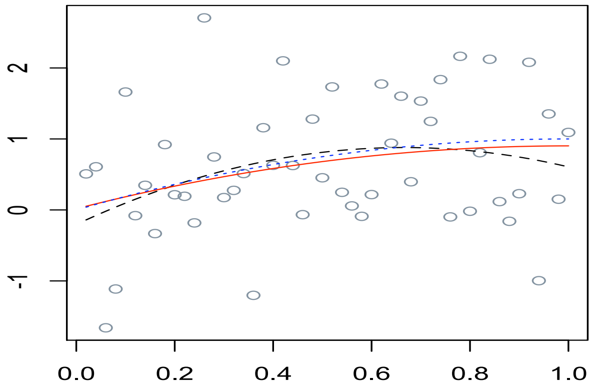

Figure 1: Quadratic fits to a scatterplot simulated using $f(x) = 1 - (1 - x)^2$ (shown as the dotted curve) and $n = 50$ observations. The dashed curve is the unconstrained fit, and the solid is constrained to be non-decreasing.

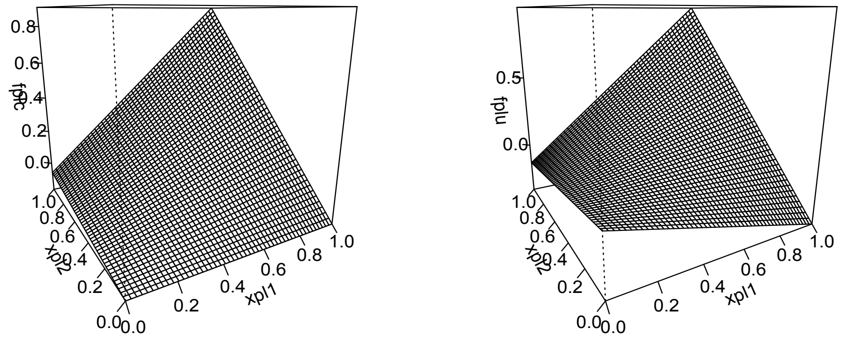

Figure 2: (a) Warped-plane fit to data set $y_i = x_1 x_2 + \epsilon_i$, where the expected value of y is constrained to be increasing in both predictors. (b) Unconstrained fit.

The $\bar{B}$ test is outlined for the transformed model $\boldsymbol{z} = \boldsymbol{\phi} + \sigma\boldsymbol{\epsilon}$, where $\boldsymbol{\epsilon}$ is multivariate standard normal, and it is known that $\boldsymbol{A\phi} \geq \boldsymbol{0}$. The test of $H_0 : \boldsymbol{A\phi} = \boldsymbol{0}$ versus $H_1 : \boldsymbol{A\phi} > \boldsymbol{0}$ uses the test statistic $\bar{B} = (SSE_0 - SSE_a)/SSE_0$, where SSE_0 is the sum of squared residuals under H_0 and SSE_a is that under H_a. Define p_d to be the probability under the null hypothesis that $\boldsymbol{z}$ falls in a sector whose face has dimension d; then

$$P(\bar{B} \leq a) = \sum_{d=0}^{m_1} P\left[Be\left(\frac{d}{2}, \frac{n-d-r}{2}\right) \leq a\right] p_d,$$

where r is the dimension of the null space of $\boldsymbol{A}$, and $Be(\alpha, \beta)$ is a beta random variable with degeneracies $Be(0, \beta) = 0$, and $Be(\alpha, 1) = 1$. See Silvapulle and Sen (2005), chapter 3 for more details. The quantities p_d may be found to arbitrary precision through simulations, which may be performed quickly using the hinge algorithm. Tests $H_0 : \boldsymbol{A}_0\boldsymbol{\theta} = \boldsymbol{d}$ versus $H_a : \boldsymbol{A}_0\boldsymbol{\theta} > \boldsymbol{d}$ for a more general quadratic programming problems (1) may be performed through transformation to cone projection.

If the constraints are valid, the test has better power when the constraints are used. For the case of simple linear regression with the slope constrained to be positive, the test of $H_0 : \beta_1 = 0$ versus $H_a : \beta_1 > 0$ is simply a one-sided t-test. For the fits in Figure 1, the F-test for $H_0 : \boldsymbol{A\beta} = 0$ (i.e., constant function) vs $H_a : \boldsymbol{A\beta} \neq \boldsymbol{0}$ provides $p = .097$, so that we would conclude at $\alpha = .05$ that there is no evidence of a relationship between the response and the predictor. The $\bar{B}$ test for the constrained fit gives $p = .037$. If the researcher may also assume that the relationship is concave, the constraint $\beta_2 < 0$ is added, and the associated $\bar{B}$ test with the more stringent assumptions provides $p = .031$ (the fit is the same as the monotone fit, as this happens to be concave). Table 2 shows that the constrained test gives substantially higher power for this example, compared to the ordinary F test of constant versus quadratic.

For the “warped plane” example of Figure 2, the the $\bar{B}$ test for constant versus doubly-monotone warped plane provides a p-value of .031, which the F-test with unconstrained warped plane alternative provides a p-value of .047. Power simulations are summarized in Table 3, for

			$\alpha = 0.01$		$\alpha = 0.05$	
n	$f(x)$	σ	$\bar{B}$ test	F-test	$\bar{B}$ test	F-test
50	$1-(x-1)^2$	1	.368	.196	.647	.417
50	$1-(x-1)^2$	2	.091	.040	.264	.133
50	$1-(x-1)^2$	4	.034	.016	.125	.069
100	$1-(x-1)^2$	1	.712	.511	.897	.745
100	$1-(x-1)^2$	2	.190	.090	.426	.239
100	$1-(x-1)^2$	4	.054	.024	.179	.092

Table 2: Power comparisons for the $\bar{B}$ test and the F-test with $H_0 : f \equiv c$, each for 10,000 simulated data sets. The alternative for the $\bar{B}$ test is that f is an increasing and concave quadratic function; for the F test the alternative is unconstrained quadratic.

two underlying regression functions, two sample sizes, and three choices of model variance.

Monotone Regression

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

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

PAVA is a well-known method for finding this solution; see Silvapulle and Sen (2005), p 47 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, and hence the necessary CPU time to compute the estimator is much smaller than for PAVA.

The $\boldsymbol{\delta}$-vectors can be written as: $\delta_i^j = j - n$, for $i = 1, \ldots, j$, and $\delta_i^j = j$, for $i = j+1, \ldots, n$, for $j = 1, \ldots, m$, and V is spanned by the one-vector. Suppose at some iteration we have $J_k = \{j_1, \ldots, j_p\}$, where the elements of J_k are ordered so that $j_1 < \cdots < j_p$. Then it is easily

			$\alpha = 0.01$		$\alpha = 0.05$	
n	μ	σ	$\bar{B}$ test	F-test	$\bar{B}$ test	F-test
100	x_1x_2	1.0	.400	.271	.661	.512
100	x_1x_2	2.0	.083	.044	.237	.147
100	x_1x_2	4.0	.029	.018	.116	.072
100	$(x_1 + x_2)/2$	1.0	.355	.208	.625	.429
100	$(x_1 + x_2)/2$	2.0	.080	.034	.232	.125
100	$(x_1 + x_2)/2$	4.0	.030	.016	.117	.068
400	x_1x_2	1.0	.970	.935	.995	.983
400	x_1x_2	2.0	.363	.252	.619	.466
400	x_1x_2	4.0	.077	.042	.228	.139
400	$(x_1 + x_2)/2$	1.0	.948	.877	.989	.962
400	$(x_1 + x_2)/2$	2.0	.329	.193	.597	.403
400	$(x_1 + x_2)/2$	4.0	.079	.039	.226	.121

Table 3: Power comparisons for the $\bar{B}$ test and the F-test, each for 10,000 simulated data sets.

seen that for $1 \leq l \leq p$,

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

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 $\boldsymbol{\delta}^l$ remains positive after any subsequent additions of edges, and a similar argument can be applied to coefficients of those edges, 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 $\boldsymbol{y}$ onto the face $\mathcal{F}_J$ is easily found by averaging the y-values between the $j \in J$.

Shape-Restricted Regression Splines

The monotone regression estimator is a step function, and the convex regression estimator is piecewise linear; neither is satisfactory if f known to be smooth. A method for smoothed monotone regression using I-splines was given by Ramsay (1988) using regression splines, and Meyer (2008) extended the method to convex restrictions using C-splines. An alternative method for either is outlined here using the more familiar B-splines (see DeBoor 2001 for details). Given a set of k distinct knots over the range of the x values, a set of m piecewise degree-p spline basis functions are be defined, that span the space of such piece-wise polynomials. The spline basis vectors contain the values of these functions at the observed x_i; let the columns of the matrix $\boldsymbol{X}$ contain the basis vectors. The unconstrained fit is obtained by minimizing $\| \boldsymbol{y} - \boldsymbol{X\beta} \|^2$ over $\beta \in I\!R^m$.

A quadratic ($p = 2$) spline function is increasing if and only if it is increasing at the knots; hence if A_{ij} is the slope of the jth spline basis function at the ith knot, the monotone spline estimator minimizes the sum of squared residuals subject to $\boldsymbol{A\beta} \geq \boldsymbol{0}$. Similarly, a cubic spline function is convex if and only if it is convex at the knots, and the cubic B-spline basis functions may be used with A_{ij} equal to the second derivative of the jth spline basis function at the ith knot. Examples of constrained fits to a scatterplot are shown in Figure 3. In plot (a), the unsmoothed monotone regression is shown as the piecewise constant function, and the monotone quadratic spline with four interior knots is shown as the dot-dash curve. In plot (b), the unsmoothed convex regression function is shown as the solid piecewise linear function, and the convex cubic regression spline is the dot-dash curve. The tests for constant versus increasing and linear versus convex regression function with smoothed alternatives use the $\bar{B}$ test statistic; for simulations comparing the test to an F-test, see Meyer (2008). For either monotone or convex constraints, we can demonstrate that when the constraints hold, the constrained version of the estimator provides smaller squared error loss, by an argument similar to that for the parametric case.

Weighted and Iteratively Re-weighted Least Squares

Estimation and testing for the regression models (4) or (5) with constraints may be accom-

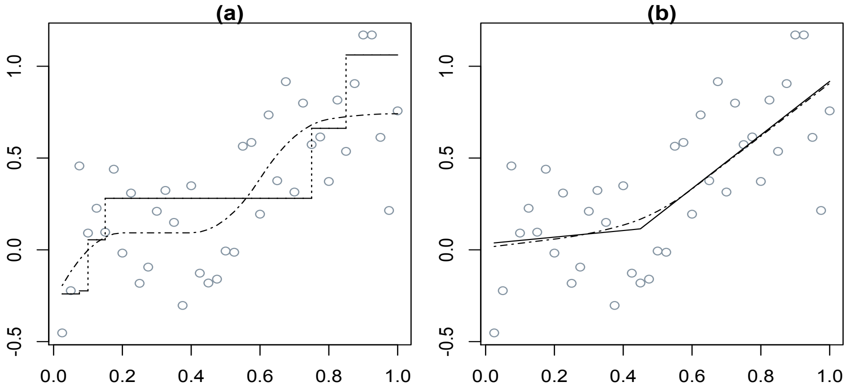

Figure 3: (a) Unsmoothed least-squares monotone fits a data set (step function) and quadratic monotone regression spline fit (dot-dash curve). (b) Least-squares convex fit (solid) and cubic convex spline (dot-dash). Both spline fits use three interior knots.

plished when the errors have an arbitrary positive definite covariance matrix, through weighted regression. Specifically, for $\boldsymbol{y} = \boldsymbol{\theta} + \sigma\boldsymbol{\epsilon}$ with $\boldsymbol{A\theta} \geq \boldsymbol{0}$, suppose $\text{cov}(\boldsymbol{\epsilon}) = \boldsymbol{\Sigma}$. The model equation is multiplied through by $\boldsymbol{L}^{-1}$, where $\boldsymbol{LL}' = \boldsymbol{\Sigma}$, to get $\boldsymbol{z} = \boldsymbol{\phi} + \sigma\boldsymbol{\xi}$ where $\boldsymbol{z} = \boldsymbol{L}^{-1}\boldsymbol{y}$, $\boldsymbol{\phi} = \boldsymbol{L}^{-1}\boldsymbol{\theta}$, and $\text{cov}(\boldsymbol{\xi})$ is the identity matrix. The new constraint matrix is $\boldsymbol{AL}$. The projection and inference about the regression function are accomplished using the transformed model.

More general constrained maximum-likelihood estimation problems may be solved through iteratively re-weighted cone projections. For example, we consider a generalized regression model, where the response is a vector $\boldsymbol{y}$ of independent observations from a distribution written in the form of an exponential family:

$$f(y_i) = \exp\{[y_i\theta_i - b(\theta_i)]/\tau^2 - c(y_i, \tau)\},$$

where the specifications of b and c determine the sub-family of models. Common examples are $b(\theta) = \log(1+e^\theta)$ for the Bernoulli and $b(\theta) = \exp(\theta)$ for the Poisson model. The log-likelihood function

$$\ell(\boldsymbol{\theta}, \tau) = \sum_{i=1}^{n} \left[\frac{y_i\theta_i - b(\theta_i)}{\tau^2} \right]$$

is to be maximized over appropriate constraints on $\boldsymbol{\theta}$.

The vector $\boldsymbol{\mu} = E(\boldsymbol{y})$, can be seen to be $\mu_i = b'(\theta_i)$, and the variance of y_i is $b''(\theta_i)\tau$. The mean vector is related to the predictor variables through a link function $g(\mu_i) = \eta_i$. If $\boldsymbol{\eta} = \boldsymbol{X}\boldsymbol{\beta}$, constraints of the form $\boldsymbol{A}\boldsymbol{\beta} \geq \boldsymbol{d}$ may be considered. Further, if the link function is one-to-one, then monotonicity constraints imposed on the η values also constrains the mean function to be monotone. Constrained regression splines may be used to model $\boldsymbol{\eta}$, under assumptions of monotonicity and smoothness. In any of these applications, the log-likelihood function is maximized over $\boldsymbol{\eta} \in \mathcal{C}$ where $\mathcal{C}$ is of the form (6). The algorithm involves iteratively re-weighted cone projections, and follows the same ideas for the generalized linear model as found in McCullagh & Nelder (1989). Starting with $\boldsymbol{\eta}^0 \in \mathcal{C}$, the estimate $\boldsymbol{\eta}^{k+1}$ is obtained from $\boldsymbol{\eta}^k$ by constructing $\boldsymbol{z}$

$$z_i = \eta_i^k + (y_i - \mu_i^k)\left(\frac{d\eta}{d\mu}\right)_{ki},$$

where $\mu_i^k = g^{-1}(\eta_i^k)$ and the derivative of the link function is evaluated at μ_i^k. The weighted projection of $\boldsymbol{z}$ onto $\mathcal{C}$ is obtained with weight vector $\boldsymbol{w}$, where $1/w_i^k = (d\eta/d\mu)_k^2 V_k$, and V_k is the variance function evaluated at μ_i^k. (The variance function may be written in terms of the mean as $V(\mu_i)$.) This scheme can be shown to converge to the value of $\boldsymbol{\eta}$ that maximizes ℓ over $\boldsymbol{\eta} \in \mathcal{C}$.

To illustrate, we use a data set given in Ruppert, Wand, and Carroll (2003), concerning incidence of bronchopulmonary dysplasia in $n = 223$ low birth weight infants. Suppose experts believe that the probability of the condition is a smooth, decreasing function of birth weight. The data are shown in Figure 4 as tick marks at one for infants the condition and at zero otherwise. The ordinary logistic regression is shown as the dashed curve, but perhaps there is no reason to believe that the assumption of linear log-odds holds. The unconstrained spline estimator with five equally spaced knots is shown as the dotted curve; this is unsatisfactory because it violates the assumption of decreasing probability. The constrained spline estimator using the same knots is shown as the solid curve; this shows a steeper descent than the fit

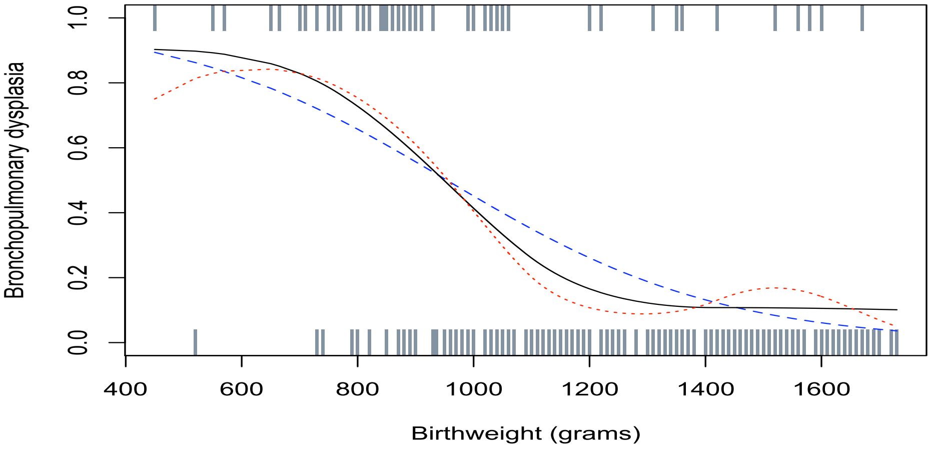

Figure 4: The estimated probability of bronchopulmonary dysplasia in low birth weight infants, as a function of birthweight. The dashed curve is the ordinary logistic regression, the dotted curve is the unconstrained regression spline estimate, and the solid curve is the monotone decreasing regression spline estimate.

assuming linear log odds, and a leveling off above zero. If the only valid assumptions are that the probability of the condition is a smooth, decreasing function of birthweight, the more flexible nonparametric fit may be preferred.

Discussion

Constrained estimation is useful for both parametric and non-parametric function estimation. When constraints are valid, their use reduces squared error loss of the estimates and increases power of the tests for significance of the predictor. The hinge algorithm is intuitively simple and quite fast. It provides a new, faster method for monotone regression. The code provided in Appendix A is short and uses only basic matrix operations. More specific code for the other applications listed in this paper, including the for the general quadratic programming problem, constrained parametric regression, constrained spline regression, the $\bar{B}$ test in parametric and non-parametric applications, and the generalized regression models, can be found on `http://www.stat.colostate.edu/∼meyer/code.htm`.

Acknowledgments

The author would like to thank two referees and an associate editor for helpful comments. This work was partially supported by NSF DMS 0905656.

A Code

The hinge algorithm to solve the generic cone projection problem of minimizing $\| \boldsymbol{y} - \boldsymbol{\theta} \|^2$ subject to $\boldsymbol{A}\boldsymbol{\theta} \geq \mathbf{0}$ is provided here in `R` code, where `amat` is the irreducible constraint matrix $\boldsymbol{A}$. Specifically, $\boldsymbol{y}$ is projected onto the polar cone to get $\hat{\boldsymbol{\rho}}$, and Proposition 2 is used to get $\hat{\boldsymbol{\theta}}$.

```
quadprog=function(y,amat){
    n=length(y);m=length(amat)/n
    sm=1e-8;h=1:m<0;obs=1:m;check=0
    delta=-amat;b2=delta%*%y
    if(max(b2)>sm){
        i=min(obs[b2==max(b2)])
        h[i]=TRUE
    }else{check=1;theta=1:n*0}
    while(check==0){
        xmat=matrix(delta[h,],ncol=n)
        a=solve(xmat%*%t(xmat))%*%xmat%*%y
        if(min(a)<(-sm)){
            avec=1:m*0;avec[h]=a
            i=min(obs[avec==min(avec)])
            h[i]=FALSE;check=0
        }else{
            check=1
            theta=t(xmat)%*%a
            b2=delta%*%(y-theta)
            if(max(b2)>sm){
                i=min(obs[b2==max(b2)])
                h[i]=TRUE;check=0}}
        }
return(y-theta)}
```

B 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) $\| \boldsymbol{y} - \hat{\boldsymbol{\theta}} \|^2$ 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

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

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 $\langle \boldsymbol{y} - \hat{\boldsymbol{\theta}}^k, \boldsymbol{\delta}^l \rangle$, because of orthogonality:

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

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

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:

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

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:

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

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

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

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

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

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

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

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

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

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 $\parallel \boldsymbol{y} - \boldsymbol{\theta}(x) \parallel^2$, we have

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

Further,

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

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

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

References

[1] Bartolucci, F. and Forcina A. (2000) A Likelihood Ratio Test for MTP$_2$ within Binary Variables. *Annals of Mathematical Statistics* **28(4)**, 1206-1218.

[2] Bartolucci, F., Forcina A., and Dardanoni, V. (2001) Positive Quadrant Dependence and Marginal Modeling in two-Way Table with Ordered Margins. *Journal of the American Statistical Association* **96** 1497-1505.

[3] de Boor, C. (2001) *A Practical Guide to Splines, revised edition.* Springer, New York.

[4] Brunk, (1955) Maximum likelihood estimates of monotone parameters. *Annals of Mathematical Statistics* **26(4)**, 607-616.

[5] Dardanoni, V., and Forcina, A. (1998) A Unified Approach to Likelihood Inference on Stochastic Orderings in a Nonparametric context. *Journal of the American Statistical Association* **93** 1112-1123.

[6] Dykstra, R.J. (1983) An Algorithm for Restricted Least Squares Regression. *Journal of the American Statistical Association* **78** 837-842.

[7] Fang, S.C. and Puthenpura, S. (1993) *Linear Optimization and Extensions. Theory and Algorithms.* Prentice Hall, Englewood Cliffs, New Jersey.

[8] 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

[9] Gourieroux, C., Holly A., and Monfort A. (1982) Likelihood Ratio Test, Wald Test, and Kuhn-Tucker Test in Linear Models with Inequality Constraints on the Regression Parameters. *Econometrica* **50(1)**, 63-80.

[10] Gourieroux, C. and Monfort A. (1995) Statistics and Econometric Models. Cambridge University Press.

[11] Hall D.B. and Præstgaard (2001) Order-Restricted Score Tests for Homogeneity in Generalised Linear and Nonlinear Mixed Models. *Biometrika* **88(3)**739-751.

[12] Hawkins, D.M. (1994) Fitting Monotonic Polynomials to Data. *Computational Statistics* **9** 233-247.

[13] Judge, G.G. and Takayama, T. (1966) Inequality Restrictions in Regression Analysis. *Journal of the American Statistical Association* **61** 166-181.

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

[15] Kudô, A. (1963) A multiariate analogue of the one-sided test. *Biometrika* **50(3,4)** 403-418.

[16] Lawson, C.L. and Hanson, R.J. (1974) *Solving Least Squares Problems.* Prentice Hall, Englewood Cliffs, New Jersey.

[17] Liew, C.K. (1976) Inequality Constrained Least-Squares Estimation. *Journal of the American Statistical Association* **71** 746-751.

[18] McCullagh, P. and Nelder, J.A. (1989) *Generalized Linear Models, Second Edition* Chapman & Hall, New York.

[19] Meyer, M.C. (1999) An Extension of the Mixed Primal-Dual Bases Algorithm to the Case of More Constraints than Dimensions, *Journal of Statistical Planning and Inference* **81**, pp13-31.

[20] Meyer M.C. (2003) A test for linear versus convex regression function using shape-restricted regression, *Biometrika*, **90(1)** 223-232.

[21] Meyer, M.C. (2008) Inference using Shape-Restricted Regression Splines. *Annals of Applied Statistics* **2(3)** 1013-1033.

[22] Molenberghs G., and Verbeke G. (2007) Likelihood ratio, score, and Walkd tests in a constrained parameter space. *American Statistician* **61(1)** 22-27.

[23] Ramsay, J. O. (1988) Monotone regression splines in action, *Statistical Science*, **3(4)** 425-461.

[24] Raubertas, R.F., Lee C.I.C., and Nordheim E.V. (1986) Hypothesis Tests for Normal Means Constrained by Linear Inequalities. *Communications in Statistics - Theory and Methods* **15(9)** 2809-2833.

[25] Robertson, T. and Wright, F. T. (1981) Likelihood Ratio Tests For and Against a Stochastic Ordering between Multinomial Populations. *Annals of Statistics* **9(6)** 1248-1257.

[26] Ruppert, D., Wand, M. P., and Carroll, R. J. (2003) *Semiparametric Regression*, Cambridge Series in Statistical and Probabilistic Mathematics.

[27] Silvapulle, M.J. and Sen, P.K. (2005) *Constrained Statistical Inference.* John Wiley & Sons, New York

[28] van Eeden, C. (2006) *Restricted Parameter Space Estimation Problems.* Springer, New York.

[29] Wolak, F.A. (1989) An Exact Test for Multiple Inequality and Equality Constraints in the Linear Regression Model. *Journal of the American Statistical Association* **82(399)** 782-793.