Bounded-Variable Least-Squares: an Algorithm and Applications

P.B. Stark
Department of Statistics
University of California
Berkeley CA 94720-3860
USA

R.L. Parker
Institute for Geophysics and Planetary Physics
University of California A-025
La Jolla CA 92093
USA

Summary

The Fortran subroutine BVLS (bounded variable least-squares) solves linear least-squares problems with upper and lower bounds on the variables, using an active set strategy. The unconstrained least-squares problems for each candidate set of free variables are solved using the QR decomposition. BVLS has a "warm-start" feature permitting some of the variables to be initialized at their upper or lower bounds, which speeds the solution of a sequence of related problems. Such sequences of problems arise, for example, when BVLS is used to find bounds on linear functionals of a model constrained to satisfy, in an approximate l_p-norm sense, a set of linear equality constraints in addition to upper and lower bounds. We show how to use BVLS to solve that problem when $p = 1, 2$, or ∞, and to solve minimum l_1 and l_∞ fitting problems. FORTRAN 77 code implementing BVLS is available from the `statlib gopher` at Carnegie Mellon University.

Keywords: Optimization, constrained least-squares, l_1 and l_∞ regression.

Acknowledgments: The first author received support from the NSF through PYI award DMS-8957573 and grant DMS-9404276, and from NASA through grant NAGW 2516.

1 Introduction

In our research we have often encountered linear least-squares, l_1, and l_∞ regression problems with linear inequality constraints on the unknowns, as well as the problem of finding bounds on linear functionals subject to upper and lower bounds on the variables and a bound on the l_1, l_2, or l_∞ misfit to a set of linear relations ([7, 8, 13, 16, 14, 12, 6]). We have used the NNLS (non-negative least-squares) algorithm of Lawson and Hanson [5] to solve some of these problems, and in principle, NNLS can solve them all. However, in practice, when either (1) both lower and upper bounds on the variables are given, or (2) one must solve a sequence of related problems, NNLS can be impractical. In (1), the size of the matrix and the number of unknowns for NNLS is unnecessarily large, since many slack variables are needed. In (2), NNLS incurs high overhead in finding a good free set from scratch in each problem (it overwrites the information needed for warm starts). We have been able to solve much larger problems in much less time using an algorithm that explicitly incorporates upper and lower bounds on the variables and returns information about its final free and bound sets.

BVLS (bounded-variable least-squares) is modelled on NNLS and solves the problem *bvls*:

$$\min_{l \le x \le u} \|Ax - b\|_2 \tag{1}$$

where $l, x, u \in \mathbf{R}^n$, $b \in \mathbf{R}^m$, and A is an m by n matrix. The relative size of m and n is immaterial; typically, in inverse problems, $m << n$. Some commercial codes advertised to work when $m < n$ do not.

As we show below, if one can solve problem *bvls* one can also solve the problem *bvmm(p)* (bounded-variable minimum misfit):

$$\min_{l \le x \le u} \|Ax - d\|_p \tag{2}$$

when $p = 1$ or $p = \infty$ ($p = 2$ is just *bvls*); as well as the problem *blf(p)*:

$$\min c \cdot x \tag{3}$$

subject to the constraints

$$l \le x \le u \tag{4}$$

and

$$\|Ax - d\|_p \le \chi \tag{5}$$

where p is any of $1, 2,$ or ∞.

2 The BVLS Algorithm

BVLS uses an active set strategy similar to that of NNLS [5], except two active sets are maintained—one for variables at their lower bounds and one for variables at their upper bounds. The proof that BVLS converges to a solution of problem *bvls* follows that for NNLS [5]; we do not give it here.

Here is an outline of the algorithm, numbered to agree with Lawson and Hanson's notation. The set $\mathcal{F}$ contains the indices of "free" components of the working solution x that are strictly between their lower and upper bounds; $\mathcal{L}$ contains the indices of components at their lower bounds; and $\mathcal{U}$ contains the indices of components at their upper bounds.

1. If this is a warm start, $\mathcal{F}$, $\mathcal{L}$ and $\mathcal{U}$ were initialized externally; set each component of x whose index is in $\mathcal{F}$ to the average of its lower and upper bounds, and set each component of x whose index is in $\mathcal{L}$ and $\mathcal{U}$ to its corresponding bound. If this is a warm start and $\mathcal{F} = \{1, \cdots, n\}$, stop with an error message. If this is a cold start, set $\mathcal{F} = \mathcal{U} = \emptyset$, $\mathcal{L} = \{1, \cdots, n\}$, and set every element of x to its lower bound.

2. Compute $w = A^T(b - Ax)$, the negative of the gradient of the squared objective.

3. If $\mathcal{F} = \{1, \cdots, n\}$, or, if $w_j \leq 0$ for all $j \in \mathcal{L}$ and $w_j \geq 0$ for all $j \in \mathcal{U}$, go to step 12. (This is the Kuhn-Tucker test for convergence.)

4. Find $t^* = \arg\max_{t \in \mathcal{L} \cup \mathcal{U}} s_t w_t$, where $s_t = 1$ if $t \in \mathcal{L}$ and $s_t = -1$ if $t \in \mathcal{U}$. If t^* is not unique, break ties arbitrarily.

5. Move t^* to the set $\mathcal{F}$.

6. Let b' be the data vector less the predictions of the bound variables; *i.e.*, $b'_j = b_j - \sum_{k \in \mathcal{L} \cup \mathcal{U}} A_{jk} x_j$. Let A' be the matrix composed of those columns of A whose indices are in $\mathcal{F}$. Let j' denote the index of the column of A' corresponding to the original index $j \in \mathcal{F}$. Find $z = \arg\min \|A'z - b'\|_2^2$.

7. If $l_j < z_{j'} < u_j$ for all j', set $x_j = z_{j'}$ and go to step 2.

8. Let $\mathcal{J}$ be the set of indices of components of z that are out-of-bounds. Define
$$q' = \arg \min_{j' \in \mathcal{J}} \min \left\{ \left| \frac{l_j - x_j}{z_{j'} - x_j} \right|, \left| \frac{u_j - x_j}{z_{j'} - x_j} \right| \right\}.$$

9. Set
$$\alpha = \min \left\{ \left| \frac{l_q - x_q}{z_{q'} - x_q} \right|, \left| \frac{u_q - x_q}{z_{q'} - x_q} \right| \right\}.$$

10. Set $x_j := x_j + \alpha(z_{j'} - x_j)$ for all $j \in \mathcal{F}$.

11. Move to $\mathcal{L}$ the index of every component of x at or below its lower bound. Move to $\mathcal{U}$ the index of every component at or above its upper bound. Go to step 6.

12. Done.

FORTRAN 77 code implementing BVLS is available through the `statlib` on-line software library at Carnegie-Mellon University. The code includes a number of features to enhance numerical stability that are not evident in the outline above. These features are described below.

As noted by Lawson and Hanson, roundoff errors in computing w may cause a component on a bound to appear to want to become free, yet when the component is added to the free set, it moves away from the feasible region. When that occurs, the component is not freed, the corresponding component of w is set to zero, and the program returns to step 3 (the Kuhn-Tucker test).

When the solution of the unconstrained least-squares problem in step 6 is infeasible, it is used in a convex interpolation with the previous solution to obtain a feasible vector (as in NNLS) in step 11. Using a convex combination ensures that the value of the objective function is lower at the new iterate (the step is taken in a descent direction). The constant in this interpolation is computed to put at least one component of the new iterate x on a bound. However, because of roundoff, sometimes no interpolated component ends up on a bound. Then in step 11, the component that determined the interpolation constant in step 8 is forced to move to the appropriate bound. This guarantees that "Loop B" (Lawson and Hanson's labelling) is finite. Also following Lawson and Hanson, any component remaining infeasible at the end of step 11 is moved to its nearer bound.

There are several differences between NNLS and BVLS that improve numerical stability: Our implementation of BVLS uses the QR decomposition to solve the unconstrained least-squares problem in step 6, as does NNLS. While NNLS updates the QR decomposition each time step 6 is entered, for efficiency, we compute the decomposition from scratch each time step 6 is executed, for stability. We also assume the solution is essentially optimal if the norm of the residual vector is less than a small fraction of the norm of the original data vector b (10^{-12} in the code below); this test might be removed by the more cautious.

If the columns of A passed to the QR routine at step 6 are linearly dependent, the new component is not moved from its bound: the corresponding component of w is set to zero, and control returns to step 3, the Kuhn-Tucker test. When the columns of A are nearly linearly dependent, we have observed cycling of free components: a component just moved to a bound tries immediately to become free; the least-squares step (6) returns a feasible value and a different component is bound. This component immediately tries to become free again, and the original component is moved back to its previous bound. We have taken two steps to avoid this problem. First, the column of the matrix A corresponding to the new potentially free component is passed to QR as the last column of its matrix. This ordering tends to make a component recently moved to a bound fail the "round-off" test mentioned above. Second, we have incorporated a test that prohibits short cycles. If the most recent successful change to the free set was to bind a particular component, that component can not be the next to be freed. This test occurs just after the Kuhn-Tucker test (step 3).

3 Using BVLS to Solve Minimum l_1 and l_∞ Problems

Consider the problem *bvmm(p)* (bounded-variable minimum misfit in the p norm):

$$\min_{l \le x \le u} \|Ax - b\|_p. \tag{6}$$

The problem *bvls* is the case $p = 2$. The cases $p = 1$ and $p = \infty$ can be written as linear programs, but can also be solved iteratively using algorithm BVLS. For reference, we give the linear programming formulations of these problems.

Problem *bvmm(1)* is solved by the following linear program:

$$\min \sum_{j=1}^{m} (s_j + t_j) \tag{7}$$

subject to the inequality constraints

$$l \le x \le u, \quad 0 \le s, \quad \text{and} \quad 0 \le t, \tag{8}$$

where $x \in \mathbf{R}^n$, and $s, t \in \mathbf{R}^m$; and the linear equality constraints

$$Ax - b + s - t = 0. \tag{9}$$

The proof of the equivalence of this linear program to *bvmm(1)* is just the observation that $\|s - t\|_1 \le \sum_j (s_j + t_j)$, and that whatever $s_j - t_j$ may be, equality is possible without changing the fit to the data or violating the inequality constraints.

Problem *bvmm(∞)* is solved by the linear program:

$$\min r \tag{10}$$

subject to the linear inequality constraints

$$r \ge 0, \quad l \le x \le u, \quad \text{and} \quad -\mathbf{1}r \le s \le \mathbf{1}r, \tag{11}$$

where $r \in \mathbf{R}$, $x \in \mathbf{R}^n$, $s \in \mathbf{R}^m$, and $\mathbf{1}$ is an m-vector of ones; and the linear equality constraints

$$Ax - b + s = 0. \tag{12}$$

3.1 Solving Problem *bvmm(1)* With BVLS

A computationally effective strategy is to mimic the linear program above, using large weights to impose the equality constraints. Let $\tilde{A}$ be the matrix A with its rows renormalized to unit Euclidean norm (to improve numerical stability), and let $\tilde{b}$ be the vector b scaled by the same constants as the rows of $\tilde{A}$. Define the matrix

$$G \equiv \begin{bmatrix} \tilde{A} & I & -I \\ \mathbf{0} & \gamma\mathbf{1} & \gamma\mathbf{1} \end{bmatrix}, \tag{13}$$

where I is an m by m identity matrix, $\mathbf{0}$ is a row vector of n zeros, $\mathbf{1}$ is a row-vector of m ones, and $\gamma << 1$ is a positive scalar discussed later. Define the $m+1$-vector

$$d \equiv \begin{bmatrix} \tilde{b} \\ 0 \end{bmatrix}, \tag{14}$$

and the $n+2m$-vectors

$$e \equiv \begin{bmatrix} l \\ 0 \end{bmatrix}, \quad \text{and} \quad f \equiv \begin{bmatrix} u \\ \text{“}\infty\text{”} \end{bmatrix}. \tag{15}$$

Here "∞" is a very large number. Now consider the *bvls* problem

$$\min \|Gz - d\|_2 \;\text{ such that }\; e \le z \le f, \tag{16}$$

where $z \in \mathbf{R}^{n+2m}$. If γ is sufficiently small, the equations involving the matrix A will be satisfied with high accuracy, while the least-squares misfit will derive almost entirely from the last row, which is the l_1 misfit. A different approach is given below in section on 4.1.

3.2 Solving Problem *bvmm(∞)* With BVLS

The strategy we employ to solve *bvmm(∞)* as a sequence of *bvls* problems is to find the largest constant r such that constraining the infinity-norm misfit to be at most r still yields a feasible problem. In implementations, we have used a bisection search to find the smallest the smallest positive value of r such that the *bvls* problem

$$\min_{l \le x \le u,\; -r\mathbf{1} \le s \le r\mathbf{1}} \left\| \begin{bmatrix} A & I \end{bmatrix} \cdot \begin{bmatrix} x \\ s \end{bmatrix} - b \right\|_2^2 \approx 0, \tag{17}$$

(within machine precision) where $\mathbf{1}$ is an m-vector of ones. This strategy is quite stable numerically and less demanding on memory than introducing additional slack variables for each data relation. Numerical stability can be further improved by normalizing each row of A to unit Euclidean length, and adjusting the constraints on the vector s accordingly. The number of iterations in the search for r can be reduced substituting for the bisection an algorithm that uses derivatives. It is clear from set inclusions that the *bvls* misfit is a monotonic function of r, so the searches are straightforward.

In all of *bvmm(p)*, $p = 1, 2, \infty$, it is trivial to incorporate weights on the data to account for unequal error variances.

4 Solving Problem *blf(p)* With BVLS

The problems *blf(p)* (bounds on a linear functional, subject to a p-norm misfit contraint) with $p = 1, 2$, and ∞ can be solved by solving a sequence of related *bvls* problems. The following subsections describe strategies we have used successfully.

It is useful to have *a priori* bounds on $c \cdot x$, which we can find by ignoring the misfit constraints:

$$c^- \equiv \sum_{j:c_j \le 0} c_j u_j + \sum_{j:c_j > 0} c_j l_j \le c \cdot x \le \sum_{j:c_j \ge 0} c_j u_j + \sum_{j:c_j < 0} c_j l_j \equiv c^+. \quad (18)$$

4.1 $p = 1$

Problem *blf(1)* can be solved by standard linear programming techniques. However, we have found the following strategy to be faster and more reliable than naive simplex algorithms, such as that in [9].

One may test for feasibility, *i.e.*, the existence of an x_0 satisfying $l \le x_0 \le u$ and $\|Ax_0 - b\|_1 \le \chi$, by solving the following *bvls* problem:

Let

$$G \equiv \begin{bmatrix} A & I & -I & 0 \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & 1 \end{bmatrix}, \quad (19)$$

where the matrices I are m by n matrices with ones on the diagonal and zero elsewhere; 0 in the top row is a column vector of m zeros; $\mathbf{0}$ in the lower row is a row vector of n zeros; the first two $\mathbf{1}$'s in the lower row are row vectors of m zeros, and the last 1 is just the scalar 1.

Let d be the vector

$$d \equiv \begin{bmatrix} b \\ \chi \end{bmatrix} \quad (20)$$

and let e and f be given by

$$e \equiv \begin{bmatrix} l \\ \mathbf{0} \end{bmatrix}, \quad (21)$$

where $\mathbf{0}$ is a column vector of $2m + 1$ zeros, and

$$f \equiv \begin{bmatrix} u \\ \text{“}\infty\text{''} \\ \chi \end{bmatrix}, \quad (22)$$

where "∞" is a $2m$-vector of numbers large enough that the bounds can not be active. It is possible to compute how large these numbers need to be *a priori* from A and b; alternatively, one can use trial values and verify *a posteriori* whether the bounds are active, and increase the value if they are.

The problem *blf(1)* is feasible if the *bvls* problem

$$\min_{y \in \mathbf{R}^{n+2m+1}} \|Gy - h\|_2 \ \text{ s.t. } \ e \le y \le f \quad (23)$$

has the value zero within machine precision. Then the first n elements of the solution vector y comprise a feasible point for the problem *blf(1)*, if one exists. This also gives an alternative algorithm for solving *bvmm(1)*: as in the algorithm for *bvmm(∞)*, use a bisection search to find the smallest value of χ for which the *bvls* problem has value 0 within machine precision.

The information in the `istate` vector can be used to warm-start the iterations about to be described. Re-define

$$G \equiv \begin{bmatrix} A & I & -I & \mathbf{0} & \mathbf{0} & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \mathbf{1} & 1 & 0 & 0 \\ c & \mathbf{0} & \mathbf{0} & 0 & 1 & -1 \\ \mathbf{0} & \mathbf{0} & \mathbf{0} & 0 & \gamma & \gamma \end{bmatrix}. \tag{24}$$

In this definition, the matrices I are as before, and the vectors 0 in the top row are column-vectors of m zeros. In the second row, the first 0 is a row-vector of n zeros; the first two vectors $\mathbf{1}$ are row-vectors of m ones; and the third 1 and the last two 0's are scalars. In the third row, c is the objective n-vector (written as a row-vector); the first two $\mathbf{0}$'s are row vectors of m zeros; and the third 0, the 1 and the -1 are scalars. In the last row, the first $\mathbf{0}$ is a row-vector of n zeros; the next two $\mathbf{0}$'s are row-vectors of m zeros; and the last three entries are the scalars 0, γ and γ. The scalar γ is a small positive number used to downweight the last row relative to those above, which we want to treat as equality constraints. The dimension of G is thus $m+3$ by $n+2m+3$.

Let d now be the $m+3$-vector

$$d \equiv \begin{bmatrix} b \\ \chi \\ c^{\pm} \\ 0 \end{bmatrix}, \tag{25}$$

where c^{+} is used to maximize $c \cdot x$, and c^{-} is used to minimize $c \cdot x$ (see the definitions 18). Let e and f be given by

$$e \equiv \begin{bmatrix} l \\ 0 \end{bmatrix}, \tag{26}$$

where 0 is a column vector of $2m+3$ zeros, and

$$f \equiv \begin{bmatrix} u \\ \infty \\ \chi \\ \infty \\ \infty \end{bmatrix}, \tag{27}$$

where ∞ is a $2m$-vector of numbers large enough that the constraints will never be active, and the last two entries ∞ are two more such scalars.

Let x^* denote the n-vector consisting of the first n elements of the solution of the *bvls* problem

$$\arg\min \{\|Gz - d\| : e \le z \le f\}. \tag{28}$$

If γ is chosen well, $c \cdot x^*$ gives the optimal value of *blf(1)*. One should verify *a posteriori* that the l_1 misfit $\|Ax^* - b\|_1$ is adequately close to χ; if not, γ needs to be adjusted. A proof that this scheme works follows that suggested in the next section for *blf(2)*.

We remark that the dimension of the matrix G is larger than it needs to be; it is possible to eliminate the last row and last two columns, substituting the row $[\gamma c\ 0\ 0\ 0]$ for $[c\ 0\ 0\ 0\ 1\ 1]$ and making corresponding changes to d, e, and f. However, we have found the scheme spelt out above to be more stable and less sensitive to the choice of γ.

4.2 $p = 2$

This problem can be solved relatively easily using BVLS and a scheme that lies conceptually between the Lagrangian dual and the use of linear constraints directly. (See Rust and Burris [10] for a different algorithm.) The basic idea is to minimize the misfit to the data subject to the inequality constraints on x and the constraint $c \cdot x = \gamma$. The value of γ is varied from the "best-fitting" value to determine the range of values for which the minimum misfit is at most χ.

We can find the "best-fitting" value of γ by solving the *bvls* problem

$$x_0 = \arg \min_{l \le x \le u} \|Ax - b\|_2^2, \tag{29}$$

and setting $\gamma_0 \equiv c \cdot x_0$.

Consider finding $\min c \cdot x$; the case for the maximum follows by changing the sign of c or other obvious modifications. Define the matrix

$$G \equiv \begin{bmatrix} A \\ \alpha c \end{bmatrix} \tag{30}$$

and the vector

$$d \equiv \begin{bmatrix} b \\ \alpha\gamma \end{bmatrix}. \tag{31}$$

Now consider

$$\Phi(\gamma) \equiv \min_{l \le x \le u} \|Gx - d\|_2^2, \tag{32}$$

and let $x^*(\gamma)$ denote the minimizer. This is a *bvls* problem. We assert that for $\gamma_t < \gamma_0$, $c \cdot x^*(\gamma)$ is the solution to

$$\min c \cdot x \quad \text{such that} \quad l \le x \le u \quad \text{and} \quad \|Ax - b\|_2 \le \|Ax^*(\gamma) - b\|_2. \tag{33}$$

The proof just relies on the convexity of the linear functional $c \cdot x$, the strict convexity of $\|Ax - b\|_2^2$, and the convexity of the set $l \le x \le u$. One may thus solve *blf(2)* by finding the smallest and largest values of γ such that $\|Ax^*(\gamma) - b\|_2 \le \chi$. The search is straightforward since $\|Ax^*(\gamma) - b\|$ increases monotonically with $|\gamma - \gamma_0|$ (the functional is quasiconvex in γ). If, during the search, $c \cdot x^*(\gamma) \le c^-$ (or $c \cdot x^*(\gamma) \ge c^+$) while $\|Ax^*(\gamma) - b\| < \chi$, then the minimum (maximum, respectively) value of $c \cdot x$ is c^- (c^+). The warm-start feature of BVLS affords large economies in solving the sequence of *bvls* problems for different values of γ, especially since the sets of components at their upper and lower bounds change gradually as γ varies.

The positive constant α is chosen to enhance numerical stability; we have found that for n up to about 200, $\alpha \approx 10^{-3}$ works well in double precision on 32 bit machines. One should allow some tolerance in attaining the exact value of χ (*e.g.,* 1% of the nominal value) to avoid excessive searching. This may be justified by remembering that in practice one picks χ using (uncertain) estimates of the standard deviations of the data errors.

4.3 $p = \infty$

This problem can also be solved directly using linear programming; we have found the following approach using BVLS to be faster and more stable than simple simplex methods. Reducing problem *blf*(∞) to a *bvls* problems is similar to the treatment for $p = 1$.

As noted in section 3.2, to find a feasible point one may solve

$$\arg \min_{e \le z \le f} \|Gz - d\|_2^2, \tag{34}$$

where

$$G \equiv \left[\begin{array}{ccc} \Lambda A & I & \mathbf{1} \end{array} \right], \tag{35}$$

(Λ is a diagonal matrix introduced to improve numerical stability; its entries are the reciprocals of the Euclidean norms of the corresponding rows of A; I is an m by m identity matrix, and $\mathbf{1}$ is a column-vector of n ones)

$$e \equiv \left[\begin{array}{c} l \\ -\Lambda \cdot \chi \end{array} \right], \tag{36}$$

where χ is an m-vector all of whose entries are χ, and

$$f \equiv \left[\begin{array}{c} u \\ \Lambda \chi \end{array} \right]. \tag{37}$$

The first n elements of z give a feasible point for the infinity-norm problem.

Let z^* solve the *bvls* problem

$$\min \|Gz - d\|_2^2 \tag{38}$$

such that $e \leq z \leq f$, where

$$G \equiv \begin{bmatrix} \Lambda A & I & 0 & 0 \\ c & \mathbf{0} & 1 & -1 \\ \mathbf{0} & \mathbf{0} & \gamma & \gamma \end{bmatrix}, \tag{39}$$

$$e \equiv \begin{bmatrix} l \\ -\Lambda \cdot \chi \\ 0 \\ 0 \end{bmatrix}, \tag{40}$$

and

$$f \equiv \begin{bmatrix} u \\ \Lambda \cdot \chi \\ \infty \\ \infty \end{bmatrix}. \tag{41}$$

Let x^* denote the n-vector composed of the first n elements of z^*. If α is chosen well, $c \cdot z^*$ solves *blf*(∞); again, one should check *a posteriori* to verify that $\|Ax^* - b\|_\infty$ is adequately close to χ.

5 Applications

As noted in the introduction, BVLS has been used to solve a variety of statistical problems arising in inverse problems. Stark and Parker [15] used BVLS to find a confidence region for the velocity with which seismic waves propagate in the Earth's core. The upper and lower bounds resulted from a nonlinear transformation that rendered the problem exactly linear, and from thermodynamic constraints on the monotonicity of velocity with radius in the Earth's outer core. Stark [11] reports joint work with D.L. Donoho using BVLS to study the extent to which positivity constraints permit "superresolution," recovering missing high frequencies beyond the linear "Rayleigh limit," in signal recovery problems. Parker and Zumberge [8], Ander *et al.* [1], and Zumberge *et al.* [17] used BVLS to test the hypothesis that Newton's law of gravitation was consistent with geophysical experiments designed to measure the "fifth force." In that problem, the inequality constraints arose from limits on the density of rocks. Hildebrand *et al.* [3] and Parker [6] used BVLS to find uncertainties in the direction of the Earth's paleomagnetic field from the magnetization of seamounts. The inequalities on the variables arose from limits on the magnetization of minerals. Stark [12] illustrated how BVLS can be used to solve a variety of inverse problems in gravimetry, geomagnetism and seismology, using the transformations of *bvmm* and blf given above. Genovese *et al.* [2] used BVLS to test hypotheses about the internal rotation of the Sun from observations of the splitting of eigenfrequencies of the "5-minute" solar vibrations. In that problem, the inequalities derived from physical hypotheses about changes in the rotation rate with position in the Sun. Johnson *et al.* [4] used BVLS to find bounds on the displacement within the Earth after earthquakes based on geodetic measurements. The inequalities arose from infinity-norm bounds on the misfit to geodetic measurements linearly related to the unknown displacement field.

The dimensions of the numerical problems in these applications reach several thousands of data and several hundreds of parameters. In every case, BVLS has been found to be numerically stable, and when the algorithm could be compared with others (for example, when BVLS was used to solve a linear programming problem by weighting, as described above, and could thus be compared with a standard simplex method), BVLS was found to be computationally more efficient, with run times up to 10 times shorter.

References

[1] M.E. Ander, M.A. Zumberge, T. Lautzenhiser, R.L. Parker, C.L.V. Aiken, M.R. Gorman, M.M. Nieto, P.R. Cooper, J.F. Ferguson, E. Fisher, G.A. McMechan, G. Sasagawa, J.M. Stevenson, G. Backus, A. Chave, J. Greer, P. Hammer, B.L. Hansen, J.A. Hildebrand, J.R. Kelty, C. Sidles, and J. Wirtz. Test of Newton's inverse-square law in the Greenland ice cap. *Phys. Rev. Lett.*, 62:985–988, 1989.

[2] C.R. Genovese, D.O. Gough, and P.B. Stark. Uncertainties for two-dimensional models of solar rotation from helioseismic eigenfrequency splitting. Technical Report 406, Dept. Stat. Stat. Univ. Cal. Berkeley, 1994.

[3] J.A. Hildebrand, J.M. Stevenson, P.T.C. Hammer, M.A. Zumberge, and R.L. Parker. A seafloor and sea-surface gravity survey of axial volcano. *J. Geophys. Res.*, B8:12,751–12,763, 1990.

[4] Johnson H.O., D.C. Agnew, and K. Hudnut. Extremal bounds on earthquake movement from geodetic data - application to the Landers earthquake. *Bull. Seis. Soc. Am.*, 84:660–667, 1994.

[5] C.W. Lawson and R.J. Hanson. *Solving Least Squares Problems*. John Wiley and Sons, Inc., New York, 1974.

[6] R.L. Parker. A theory of ideal bodies for seamount magnetism. *jgr*, 96:16101–16112, 1991.

[7] R.L. Parker and K. Whaler. Numerical methods for establishing solutions to the inverse problem of electromagnetic induction. *jgr*, 86:9574–9584, 1981.

[8] R.L. Parker and M.A. Zumberge. An analysis of geophysical experiments to test Newton's law of gravity. *Nature*, 342:39–31, 1989.

[9] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C.* Cambridge University Press, New York, 1988.

[10] B.W. Rust and W.R. Burrus. *Mathematical Programming and the Numerical Solution of Linear Equations.* American Elsevier Pub. Co., Inc., New York, 1972.

[11] P.B. Stark. Strict bounds and applications. In P.C. Sabatier, editor, *Some Topics on Inverse Problems*, pages 220–230. World Scientific, Singapore, 1988.

[12] P.B. Stark. Inference in infinite-dimensional inverse problems: Discretization and duality. *J. Geophys. Res.*, 97:14,055–14,082, 1992.

[13] P.B. Stark and R.L. Parker. Smooth profiles from $\tau(p)$ and $X(p)$ data. *Geophys. J. R. Astron. Soc.*, 89:997–1010, 1987.

[14] P.B. Stark and R.L. Parker. Velocity bounds from statistical estimates of $\tau(p)$ and $X(p)$. *J. Geophys. Res.*, 92:2713–2719, 1987.

[15] P.B. Stark and R.L. Parker. Correction to 'Velocity bounds from statistical estimates of $\tau(p)$ and $x(p)$' by Philip B. Stark and Robert L. Parker. *J. Geophys. Res.*, 93:13,821–13,822, 1988.

[16] P.B. Stark, R.L. Parker, G. Masters, and J.A. Orcutt. Strict bounds on seismic velocity in the spherical Earth. *J. Geophys. Res.*, 91:13,892–13,902, 1986.

[17] M. A. Zumberge, J. A. Hildebrand, J.M. Stevenson, R.L. Parker, A.D. Chave, M. E. Ander, and F. N. Spiess. Submarine measurement of the Newtonian gravitational constant. *Phys. Rev. Lett.*, 27:3051–3054, 1991.