On Algorithms for Solving Least Squares Problems under an L_1 Penalty or an L_1 Constraint

B.A. Turlach
School of Mathematics and Statistics (M019)
The University of Western Australia
35 Stirling Highway, Crawley WA 6009
Australia

Abstract

Tibshirani (1996) proposed the least absolute shrinkage and selection operator (LASSO) which estimates a vector of regression coefficients by minimising the residual sum of squares subject to a constraint (penalty) on the sum of the absolute values of the coefficient estimates. In this paper, we describe several algorithms that can be used to calculate the LASSO solution.

1 Introduction

If one uses linear regression to model observations $(x_{i1}, \ldots, x_{im}, y_i)$, $i = 1, \ldots, n$, where the x_{ij}s are the regressors and y_i the response for the ith observation, then ordinary least squares regression finds the linear combination of the x_{ij}s that minimises the residual sum of squares. However, if m is large, perhaps even $m > n$, or if the regressor variables are highly correlated, then the variances of the least-squares coefficient estimates may be unacceptably high. Standard methods for addressing this difficulty include ridge regression and, particularly in cases where a more parsimonious model is desired, subset selection.

As an alternative to standard ridge regression and subset selection techniques, Tibshirani (1996) proposed the least absolute shrinkage and selection operator (LASSO), which minimises the residual sum of squares under a constraint on the L_1-norm of coefficient vector. Thus the LASSO estimator solves the optimisation problem

$$\min_{\beta_1,\ldots,\beta_m} \quad \frac{1}{2}\sum_{i=1}^{n}\left(y_i - \sum_{j=1}^{m} x_{ij}\beta_j\right)^2 \tag{1a}$$

$$\text{subject to} \quad \sum_{j=1}^{m} |\beta_j| \le t. \tag{1b}$$

To simplify notation, define the response vector as $\boldsymbol{y} = (y_1, \ldots, y_n)'$, the $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_m)'$ and vectors $\boldsymbol{x}_j = (x_{1j}, \ldots, x_{nj})'$, $j = 1, \ldots, m$. Then we can write the $n \times m$ design matrix as $\mathbf{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m)$ and problem (1) in matrix form as follows:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} \quad f(\boldsymbol{\beta}) = \frac{1}{2}(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})'(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) \tag{2a}$$

$$\text{subject to} \quad g(\boldsymbol{\beta}) = t - \|\boldsymbol{\beta}\|_1 \ge 0. \tag{2b}$$

In the following, we will assume that the response vector $\boldsymbol{y}$ is centred ($\sum_i y_i = 0$) and the vectors $\boldsymbol{x}_j$, $j = 1, \ldots, m$, are standardised ($\sum_i x_{ij} = 0$ and $\sum_i x_{ij}^2/n = 1$ for all $j = 1, \ldots, m$). Note that due to this standardisation, the matrix $\mathbf{X}$ can have at most $m = n-1$ linear independent columns. If $m \le n-1$, then we assume that the matrix $\mathbf{X}$ has full column rank, otherwise we assume that there are at least $n-1$ columns such that the sub matrix build from these columns has full column rank.

The rest of the paper is structured as follows. Section 2 give an exact characterisation of the solutions of (2). It turns out that the solutions $\hat{\boldsymbol{\beta}}(t)$ of (2) as functions of t are piecewise linear and in Section 3 we describe an algorithm to calculate the complete solution path.

However, if the loss function $f(\cdot)$ in (2a) is replaced by another loss function, e.g. a likelihood based loss function to fit a generalised linear model (McCullagh and Nelder, 1989) under an L_1 constraint on the parameter vector, then the solution path $\hat{\boldsymbol{\beta}}(t)$ is, typically, no longer piecewise linear. In such situations it is of interest to solve (2), embedded within an iteratively reweighted least squares (IRLS), for a given t. An algorithm to solve (2) for a given t is described in Section 4.

In Section 5 we discuss the penalised optimisation problem that is equivalent to the constrained optimisation problem (2) and discuss an efficient algorithm to solve the penalised problem in Section 6. Some final comments are given in Section 7.

2 Characterisation of solutions

The Lagrangian function corresponding to problem (2) is

$$\mathcal{L}(\boldsymbol{\beta}, \lambda) = f(\boldsymbol{\beta}) - \lambda g(\boldsymbol{\beta}).$$

and, together with results from convex analysis (Rockafellar, 1970; Osborne, 1985; Clarke, 1990), it can be used to characterise solutions of (2). For problem (2), $\hat{\boldsymbol{\beta}}$ is a solution if, and only if, $\lambda \geq 0$ exists such that

$$\mathbf{X}'\hat{\boldsymbol{r}} = \lambda \boldsymbol{c}, \tag{3}$$

where $\hat{\boldsymbol{r}} = \boldsymbol{y} - \mathbf{X}'\hat{\boldsymbol{\beta}}$ and $\boldsymbol{c} = (c_1, \ldots, c_m)'$ is such that

$$c_i \begin{cases} = 1 & \text{if } \hat{\beta}_i > 0, \\ = -1 & \text{if } \hat{\beta}_i < 0, \\ \in [-1, 1] & \text{if } \hat{\beta}_i = 0. \end{cases}$$

For details see Osborne *et al.* (2000a,b) and Efron *et al.* (2004).

Note that $\|\boldsymbol{c}\|_\infty = \max_j\{|c_j|\} = 1$ and $\boldsymbol{c}'\hat{\boldsymbol{\beta}} = \|\hat{\boldsymbol{\beta}}\|_1$. Hence

$$\lambda = \|\mathbf{X}'\hat{\boldsymbol{r}}\|_\infty = \hat{\boldsymbol{r}}'\mathbf{X}\hat{\boldsymbol{\beta}}/\|\hat{\boldsymbol{\beta}}\|_1.$$

Since the columns of $\mathbf{X}$ have been standardised, the entries in the vector $\mathbf{X}'\hat{\boldsymbol{r}}$ are proportional to the empirical correlations between the residual vector $\hat{\boldsymbol{r}}$ and the regressors variables $\boldsymbol{x}_j$, $j = 1, \ldots, m$. Thus, two observation follow from (3):

(P1) at the solution $\hat{\boldsymbol{\beta}}$ of (2), a component of $\hat{\boldsymbol{\beta}}$ may be non-zero if, and only if, the absolute value of the correlation between the residual vector and the corresponding regressor variables is maximal; and

(P2) the sign of any non-zero component of $\hat{\boldsymbol{\beta}}$ must equal the sign of the correlation between the residual vector and the corresponding regressor variable.

One can also show, although this fact does not follow directly from (3), that if we vary t and consider the solution $\hat{\boldsymbol{\beta}}(t)$ of (2) as a function of t, then $\hat{\boldsymbol{\beta}}(t)$ is a piecewise linear function (Osborne *et al.*, 2000a; Efron *et al.*, 2004).

The fact that $\hat{\boldsymbol{\beta}}(t)$ is a piecewise linear function of t, together with properties (P1) and (P2), makes it possible to design algorithms that calculate the complete solution path $\hat{\boldsymbol{\beta}}$ very efficiently. Such an algorithm is described below in Section 3.

The results stated in this section can be readily extended to more general problems. For example, the right hand side of (3) is the gradient of the (loss) function $f(\boldsymbol{\beta})$, while the vector on the left is an element of the subgradient of $g(\boldsymbol{\beta})$ (Osborne, 1985). Thus, if we would replace the quadratic loss function $f(\cdot)$ by another loss function (e.g. a likelihood based loss function to fit generalised linear models under an L_1 constraint on the parameter vector, or Huber's ψ function (Rosset and Zhu, 2004) for robust regression), then the solutions of the modified problem can be characterised by an equation similar to (3), albeit with the right hand side replaced by the gradient of the new loss function. The solutions would have properties similar to (P1) and (P2) described above.

In general, however, the solution path $\hat{\boldsymbol{\beta}}(t)$ will no longer be piecewise linear in t. If one minimises the quadratic loss function under constraints on the parameter that are convex polyhedral, then the solution path will be linear in t (M R Osborne, personal communication). Rosset and Zhu (2004) discuss for which pairs of loss functions and penalty/constraint functions the solution path is piecewise linear. Some results on how to track a solution path that is not piecewise linear can be found in Rosset (2004).

3 Homotopy algorithm

Osborne *et al.* (2000a) showed that the solution $\hat{\boldsymbol{\beta}}(t)$ of (2) as a function of t is piecewise linear and describe an algorithm that calculates the complete solution path. The description of their algorithm concentrates heavily on the efficient implementation of the algorithm via a (partial) QR factorisation of the matrix $\mathbf{X}$.

Efron *et al.* (2004) introduce least angle regression and show how this technique relates to the LASSO and other recently proposed variable selection methods. Their discussion provides further insight into the LASSO and a geometrical interpretation of the piecewise linear homotopy that describes the complete solution path.

The algorithm for calculating the complete solution path described below is similar to the one discussed in Efron *et al.* (2004) but for a slight modification in the way the direction in which one moves during each iteration is calculated.

The algorithm starts at $t = 0$. Obviously, for that value the solution of (2) is $\hat{\boldsymbol{\beta}} = \mathbf{0}$, the fitted values are $\hat{\boldsymbol{\mu}} = \mathbf{0}$ and the residual vector is $\hat{\boldsymbol{r}} = \boldsymbol{y}$. As the algorithm progresses, only some components of $\boldsymbol{\beta}$ are allowed to be non-zero and we use a set, say σ, to keep track of the indices of these components. The question arises how σ should be initialised. As suggested by property (P1), the correct way to initialise

σ is to put all those indices j, $j = 1, \ldots, m$, into σ for which the absolute value of the correlation between the residual vector and the corresponding regressor variables is maximal. These calculations are described below in step 1 of the algorithm.

Now, as long as σ does not change the corresponding components of $\hat{\boldsymbol{\beta}}(t)$ change linearly and the other components remain fixed at zero. Thus, we have to address two questions. First, at which specific rate do the components of $\hat{\boldsymbol{\beta}}(t)$ change? Secondly, when, i.e. for which values of t, does σ change? The first question is answered using (3) and property (P1), whereas the second question is answered using properties (P1) and (P2).

Equation (3) and property (P1) imply that for all $j \in \sigma$ the absolute value of the correlation between $\boldsymbol{x}_j$ and the current residual vector $\hat{\boldsymbol{r}}$, which is proportional to $|\boldsymbol{x}_j'\hat{\boldsymbol{r}}|$, must be equal. Let $\bar{\beta}_j^0$, $j \in \sigma$, denote the estimated coefficients if we regress $\boldsymbol{y}$ only on $\boldsymbol{x}_j$, $j \in \sigma$, using unconstrained (ordinary) least squares, and define

$$\bar{\beta}_j(\gamma) = \hat{\beta}_j + \gamma(\bar{\beta}_j^0 - \hat{\beta}_j), \quad j \in \sigma, \quad \gamma \in [0,1]. \quad (4)$$

Now consider the vector $\bar{\boldsymbol{\beta}}(\gamma)$ that we obtain by using, for $j \in \sigma$, the $\bar{\beta}_j$ defined above and whose other components are fixed at 0. Obviously $\bar{\boldsymbol{\beta}}(0)$ is our current estimated coefficient vector $\hat{\boldsymbol{\beta}}$. If we define the corresponding residual vector

$$\bar{\boldsymbol{r}}(\gamma) = \boldsymbol{y} - \mathbf{X}\bar{\boldsymbol{\beta}}(\gamma)$$

then, obviously, $\boldsymbol{x}_j'\bar{\boldsymbol{r}}(1) = 0$ for all $j \in \sigma$ and it is easy to verify that, for all $j \in \sigma$ and $\gamma \in [0,1]$, $\boldsymbol{x}_j'\bar{\boldsymbol{r}}(\gamma)$ is linear in γ and, thus, $|\boldsymbol{x}_j'\bar{\boldsymbol{r}}(\gamma)| = C(\gamma)$ for some function $C(\cdot)$ independent of j.

It follows that equation (4) answers the first question and describes how $\hat{\boldsymbol{\beta}}(t)$ changes as long as σ does not change. We have to move into the direction of $\bar{\boldsymbol{\beta}}(1)$. Thus, in step 2 of the algorithm we regress $\boldsymbol{y}$ only on $\boldsymbol{x}_j$, $j \in \sigma$ and calculate $\bar{\boldsymbol{\beta}}(1)$ (denoted as $\bar{\boldsymbol{\beta}}^{(k+1)}$ below).

The calculations are set up such that a full step along $\boldsymbol{d} = \bar{\boldsymbol{\beta}}(1) - \hat{\boldsymbol{\beta}}$ would take us to $\bar{\boldsymbol{\beta}}(1)$ as our new estimated coefficient vector. However, typically we cannot take a full step since σ will change during this step which leads us to the second question. Essentially, σ will change during the step if either property (P1) or (P2) becomes violated.

Note that for all $j \in \sigma$, $\boldsymbol{x}_j'\bar{\boldsymbol{r}}(\gamma)$ changes linearly from $\boldsymbol{x}_j'\bar{\boldsymbol{r}}(0)$ to zero for $\gamma \in [0,1]$ and, thus, these quantities cannot change signs. It follows that property (P2) can only be violated during the move to $\bar{\boldsymbol{\beta}}(1)$ if, for some $j \in \sigma$, $\bar{\beta}_j(0)$ and $\bar{\beta}_j(1)$ have different signs. If this happens, we can only take a partial step along $\boldsymbol{d}$. In step 3 of the algorithm we calculate how far we can move along $\boldsymbol{d}$ (namely, $\gamma_2\boldsymbol{d}$, $\gamma_2 \geq 0$) before property (P2) is violated. If we end up taking a step of this length, then we have to remove one component from σ.

Property (P1) can only be violated if for some $j_0 \notin \sigma$ and $\gamma \in [0,1]$, the value of $|\boldsymbol{x}_{j_0}'\bar{\boldsymbol{r}}(\gamma)|$ becomes equal to $C(\gamma)$. However, for all $j \notin \sigma$, the quantities $\boldsymbol{x}_j'\bar{\boldsymbol{r}}(\gamma)$ also change linearly with $\gamma \in [0,1]$ and thus it is easy to calculate whether a violation of property (P1) happens as we move along $\boldsymbol{d}$. In step 3 of the algorithm we calculate how far we can move along $\boldsymbol{d}$ (namely, $\gamma_1\boldsymbol{d}$, $\gamma_1 \geq 0$) before property (P1) is violated. If we end up taking a step of this length, then we have to add one component from σ.

In step 4 of the algorithm, we take a step of appropriate length along $\boldsymbol{d}$, and update our vector of estimated parameters $\hat{\boldsymbol{\beta}}$ and our vector of fitted values $\hat{\boldsymbol{\mu}}$. Then in step 5 we update σ. To answer our second question above, note that the values of t at which σ changes are given by the L_1 norm of our current estimate for $\boldsymbol{\beta}$.

Finally, we calculate the current residual vector $\hat{\boldsymbol{r}}$ and $\boldsymbol{c} = \mathbf{X}'\hat{\boldsymbol{r}}$. If all components of $\boldsymbol{c}$ are zero, then we have either reached the unconstrained solution of (2a) (if $m - 1 \leq n$) or the vector of estimated coefficients with smallest L_1 norm such that $\boldsymbol{y} = \mathbf{X}\hat{\boldsymbol{\beta}}$ (if $m - 1 \geq n$). In this case we stop the iteration, otherwise we continue.

Hence, the complete algorithm is:

1. Set $\hat{\boldsymbol{\mu}}^{(0)} = \mathbf{0}$, $\hat{\boldsymbol{\beta}}^{(0)} = \mathbf{0}$ and $k = 0$. Calculate $\boldsymbol{c} = \mathbf{X}'\boldsymbol{y}$ and set $C = \|\boldsymbol{c}\|_\infty$. Initialise $\sigma = \{j : |c_j| = C\}$.

2. Set $\mathbf{X}_\sigma = (\cdots \boldsymbol{x}_j \cdots)_{j\in\sigma}$ and calculate

$$\bar{\boldsymbol{b}}_\sigma^{(k+1)} = (\mathbf{X}_\sigma'\mathbf{X}_\sigma)^{-1}\mathbf{X}_\sigma'\boldsymbol{y}$$
$$\bar{\boldsymbol{\mu}}^{(k+1)} = \mathbf{X}_\sigma\bar{\boldsymbol{b}}_\sigma^{(k+1)}$$
$$\boldsymbol{a} = \mathbf{X}'\left(\bar{\boldsymbol{\mu}}^{(k+1)} - \hat{\boldsymbol{\mu}}^{(k)}\right)$$

   Set $\bar{\boldsymbol{\beta}}^{(k+1)} = \mathbf{P}\begin{pmatrix}\bar{\boldsymbol{b}}_\sigma^{(k+1)} \\ \mathbf{0}\end{pmatrix}$; $\mathbf{P}$ a suitable permutation matrix.

3. Calculate the step size γ as $\gamma = \min(\gamma_1, \gamma_2, 1)$ where

$$\gamma_1 = \min_{j\notin\sigma}{}^{+}\left\{\frac{C - c_j}{C - a_j}, \frac{C + c_j}{C + a_j}\right\}$$

   and

$$\gamma_2 = \min_{j\in\sigma}{}^{+}\left\{-\frac{\hat{\beta}_j^{(k)}}{\bar{\beta}_j^{(k+1)} - \hat{\beta}_j^{(k)}}\right\}.$$

Here, $\min^+$ indicates that the minimum is taken only over positive components.

4. Use the calculated step length to update the estimated parameter and fitted values:

$$\hat{\boldsymbol{\mu}}^{(k+1)} = \hat{\boldsymbol{\mu}}^{(k)} + \gamma\left(\bar{\boldsymbol{\mu}}^{(k+1)} - \hat{\boldsymbol{\mu}}^{(k)}\right)$$
$$\hat{\boldsymbol{\beta}}^{(k+1)} = \hat{\boldsymbol{\beta}}^{(k)} + \gamma\left(\bar{\boldsymbol{\beta}}^{(k+1)} - \hat{\boldsymbol{\beta}}^{(k)}\right)$$

5. If $\gamma = \gamma_1$ then, with $\gamma_j = \min^+\left\{\frac{C-c_j}{C-a_j}, \frac{C+c_j}{C+a_j}\right\}$,

$$\sigma = \sigma \cup \{j : j \notin \sigma \text{ and } \gamma_j = \gamma_1\}$$

If $\gamma = \gamma_2$, then $\sigma = \sigma \setminus \left\{j : \hat{\beta}_j^{(k+1)} = 0\right\}$.

6. Calculate $\boldsymbol{c} = \mathbf{X}'\left(\boldsymbol{y} - \hat{\boldsymbol{\mu}}^{(k+1)}\right)$, set $C = \|\boldsymbol{c}\|_\infty$ and $k \leftarrow k+1$.
If $C = 0$ stop, otherwise return to step 2

4 Fixed t

Osborne *et al.* (2000b) propose the following algorithm which is based on a local linearisation of (2a) about a current $\boldsymbol{\beta}$.

To describe this algorithm, we introduce the following notation. Again, σ denotes a set and a component β_i of $\boldsymbol{\beta}$ may be non-zero if, and only if, $i \in \sigma$. The vector $\boldsymbol{\beta}_\sigma$ collects all the components of $\boldsymbol{\beta}$ that may be non-zero and is of length $|\sigma|$. The $m \times m$ matrix $\mathbf{P}$ is a suitable permutation matrix such that $\boldsymbol{\beta} = \mathbf{P}\begin{pmatrix}\boldsymbol{\beta}_\sigma \\ \mathbf{0}\end{pmatrix}$. Also $\boldsymbol{\theta}_\sigma = \text{sign}(\boldsymbol{\beta}_\sigma)$ denotes the vector whose components equal the sign of the corresponding components in $\boldsymbol{\beta}_\sigma$. Note that $\boldsymbol{\theta}_\sigma'\boldsymbol{\beta}_\sigma = \|\boldsymbol{\beta}_\sigma\|_1 = \|\boldsymbol{\beta}\|_1$. At any step of the algorithm $\boldsymbol{\beta}_\sigma$ has to be feasible, i.e. $\boldsymbol{\theta}_\sigma'\boldsymbol{\beta}_\sigma \leq t$.

Usually, the algorithm is started at $\boldsymbol{\beta} = \mathbf{0}$ and with σ initialised in the same manner as in step 1 of the homotopy algorithm. However, the algorithm can be started at an arbitrary $\boldsymbol{\beta}$ as long as this coefficient vector is feasible. This fact is useful if this algorithm is embedded within a loop; as one would do if one would replace the loss function $f(\cdot)$ in (2a) by some other loss function, e.g. a likelihood based loss function to fit generalised linear models under an L_1 constraint on the parameter vector (Lokhorst, 1999; Roth, 2002, 2004).

The algorithm described in Osborne *et al.* (2000b) solves at each step the following optimisation problem:

$$\min_{\boldsymbol{h}} \quad f(\boldsymbol{\beta} + \boldsymbol{h}) \tag{5a}$$
$$\text{where} \quad \boldsymbol{\theta}_\sigma'(\boldsymbol{\beta}_\sigma + \boldsymbol{h}_{\boldsymbol{\sigma}}) \leq t \tag{5b}$$
$$\text{and} \quad \boldsymbol{h} = \mathbf{P}\begin{pmatrix}\boldsymbol{h}_\sigma \\ \mathbf{0}\end{pmatrix} \tag{5c}$$

Let $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta} + \boldsymbol{h}$ be the solution of (5). We call $\tilde{\boldsymbol{\beta}}$ *sign feasible* if $\text{sign}(\tilde{\boldsymbol{\beta}}_\sigma) = \boldsymbol{\theta}_\sigma$.

If $\tilde{\boldsymbol{\beta}}$ is not sign feasible, we proceed as follows:

1. Move to the first new zero component in direction $\boldsymbol{h}$, i.e. find the smallest γ, $0 < \gamma < 1$ and corresponding $k \in \sigma$ such that $0 = \beta_k + \gamma h_k$.

2. Update σ by deleting k from it, setting $\boldsymbol{\beta} = \boldsymbol{\beta} + \gamma\boldsymbol{h}$, resetting $\boldsymbol{\beta}_\sigma$ and $\boldsymbol{\theta}_\sigma$ accordingly (they are still both feasible) and recompute $\boldsymbol{h}$ by solving (5) again.

3. Iterate until a sign feasible $\tilde{\boldsymbol{\beta}}$ is obtained.

If $\tilde{\boldsymbol{\beta}}$ is sign feasible, then we can test it for optimality. Calculate the corresponding residual vector $\tilde{\boldsymbol{r}} = \boldsymbol{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}$. If $\|\mathbf{X}'\tilde{\boldsymbol{r}}\|_\infty = 0$, then we stop. In this case we are either at the unconstrained solution (if $m - 1 \leq n$) or at a solution that interpolates $\boldsymbol{y}$ (if $m - 1 \geq n$). Otherwise, calculate

$$\tilde{\boldsymbol{v}} = \mathbf{X}'\tilde{\boldsymbol{r}}/\|\mathbf{X}'\tilde{\boldsymbol{r}}\|_\infty$$

If $|\tilde{v}_i| = 1$ for $i \in \sigma$ and $-1 \leq \tilde{v}_i \leq 1$ for $i \notin \sigma$, then $\tilde{\boldsymbol{\beta}}$ is a solution of (2). Otherwise, we proceed as follows.

1. Determine the most violated condition, i.e. find s such that $\tilde{v}_s$ has maximal absolute value.

2. Update σ by adding s to it. $\boldsymbol{\beta}_\sigma$ and $\boldsymbol{\theta}_\sigma$ are updated by appending a zero and $\text{sign}(\tilde{v}_s)$, respectively, as last elements.

3. Solve (5) and iterate.

5 Constrained *vs.* penalised estimation

The constrained problem (2) is, of course, equivalent to the penalised problem:

$$\min_{\boldsymbol{\beta}\in\mathbb{R}^m} (\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta})'(\boldsymbol{y} - \mathbf{X}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_1 \tag{6}$$

That is, for a given λ, $0 \leq \lambda < \infty$, there exists a $t \geq 0$ such that the two problems share the same solution, and vice versa.

Fu (1998) discusses an iterative algorithm that solves (6). However, this algorithm starts at $\hat{\boldsymbol{\beta}}^0$, the solution of the unconstrained problem, and is therefore not suitable for the case where $n < m$, i.e. if one has more variables than observations; a situation which is nowadays common in chemometrics, microarray applications and other situations.

However, in the $n < m$ case it is advantageous to use the penalised version. The solutions of (6) have

the same characterisation (3) as the solutions of (2). Using this characterisation, it is easy to show that if $\lambda \geq \|\mathbf{X}'\boldsymbol{y}\|_\infty$, then the solution to (6) is $\hat{\boldsymbol{\beta}} = \mathbf{0}$. Thus, even if $n < m$, it makes only sense in (6) to choose λ between 0 and $\lambda_{\max} = \|\mathbf{X}'\boldsymbol{y}\|_\infty$ and one could reparameterise the problem such that the penalty parameter is specified relative to $\lambda_{\max}$ and always between 0 or 1.

By way of contrast, for the constrained problem (2), it is only possible to specify *a priori* a sensible range for the bound t in the case $n > m$. In that case, only values of t between 0 and $\|\hat{\boldsymbol{\beta}}^0\|_1$ are of interest, since larger values of t would simply yield the unconstrained solution $\hat{\boldsymbol{\beta}}^0$.

6 Fixed λ

The algorithm of Osborne *et al.* (2000b) for solving (2) for fixed t, which is described above in Section 4, can be easily adapted to calculate the solution of (6) for a given λ.

We use the same notation as in Section 4. Namely, β_i may be non-zero if and only if $i \in \sigma$, $\mathbf{P}$ is the permutation matrix such that $\boldsymbol{\beta} = \mathbf{P}\begin{pmatrix}\boldsymbol{\beta}_\sigma \\ \mathbf{0}\end{pmatrix}$ and $\boldsymbol{\theta}_\sigma = \operatorname{sign}(\boldsymbol{\beta}_\sigma)$.

Again, the algorithm is started at $\boldsymbol{\beta} = \mathbf{0}$ and with σ initialised in the same manner as in step 1 of the homotopy algorithm. However, the algorithm can be started at an arbitrary $\boldsymbol{\beta}$. This fact is useful if this algorithm is embedded within an iteratively reweighted least squares (IRLS) loop.

Now we solve at each step the following optimisation problem:

$$\min_{\boldsymbol{h}} \qquad f(\boldsymbol{\beta} + \boldsymbol{h}) + \lambda \boldsymbol{\theta}'_\sigma(\boldsymbol{\beta}_\sigma + \boldsymbol{h}_{\boldsymbol{\sigma}}) \tag{7a}$$

$$\text{where} \qquad \boldsymbol{h} = \mathbf{P}\begin{pmatrix}\boldsymbol{h}_\sigma \\ \mathbf{0}\end{pmatrix} \tag{7b}$$

Let $\tilde{\boldsymbol{\beta}} = \boldsymbol{\beta} + \boldsymbol{h}$ be the solution of (7). We call $\tilde{\boldsymbol{\beta}}$ *sign feasible* if $\operatorname{sign}(\tilde{\boldsymbol{\beta}}_\sigma) = \boldsymbol{\theta}_\sigma$.

If $\tilde{\boldsymbol{\beta}}$ is not sign feasible, we proceed as follows:

1. Move to the first new zero component in direction $\boldsymbol{h}$, i.e. find the smallest γ, $0 < \gamma < 1$ and corresponding $k \in \sigma$ such that $0 = \beta_k + \gamma h_k$.
2. Update σ by deleting k from it, setting $\boldsymbol{\beta} = \boldsymbol{\beta} + \gamma\boldsymbol{h}$, resetting $\boldsymbol{\beta}_\sigma$ and $\boldsymbol{\theta}_\sigma$ accordingly and recompute $\boldsymbol{h}$ by solving (7) again.
3. Iterate until a sign feasible $\tilde{\boldsymbol{\beta}}$ is obtained.

If $\tilde{\boldsymbol{\beta}}$ is sign feasible, then we can test it for optimality. Calculate, with $\tilde{\boldsymbol{r}} = \boldsymbol{y} - \mathbf{X}\tilde{\boldsymbol{\beta}}$,

$$\tilde{\boldsymbol{c}} = \mathbf{X}'\tilde{\boldsymbol{r}}.$$

If $\tilde{c}_i = \operatorname{sign}(\tilde{\beta}_i)\lambda$ for $i \in \sigma$ and $-\lambda \leq \tilde{c}_i \leq \lambda$ for $i \notin \sigma$, then $\tilde{\boldsymbol{\beta}}$ is a solution of (6). Otherwise, we proceed as follows.

1. Determine the most violated condition, i.e. find s such that $\tilde{c}_s$ has maximal absolute value.
2. Update σ by adding s to it. $\boldsymbol{\beta}_\sigma$ and $\boldsymbol{\theta}_\sigma$ are updated by appending a zero and $\operatorname{sign}(\tilde{c}_s)$, respectively, as last elements.
3. Solve (7) and iterate.

7 Concluding remarks

The presented algorithms are specifically designed to solve (2) or (6) and are very efficient in doing so. They can be easily adapted to problems where the quadratic loss function is replaced by another loss function by embedding them within an IRLS loop (Lokhorst, 1999; Roth, 2002, 2004).

However, other algorithms have been proposed. In particular, in the wavelet literature Chen *et al.* (1999) and Sardy *et al.* (2000) use interior point algorithms to solve LASSO problems (i.e. either (2) or (6)). An interesting feature of these algorithms is that they do not construct the design matrix $\mathbf{X}$ explicitly. All necessary calculations are done via the fast wavelet transform.

By way of contrast, all the available implementations of the algorithms described in Osborne *et al.* (2000a,b) and Efron *et al.* (2004) construct the design matrix $\mathbf{X}$ explicitly and work with a (partial) QR factorisation of this matrix to perform all necessary computations fast and efficiently.

Other algorithms that have been proposed to solve LASSO problems are special cases of generalisations that allow either a more general penalty or a more general loss function (Fu, 1998; Fan and Li, 2001). Finally, Grandvalet (1998) and Grandvalet and Canu (1999) propose to use adaptive ridge regression procedures to solve LASSO problems.

References

Chen, S.S., Donoho, D.L. and Saunders, M.A. (1999). Atomic decomposition by basis pursuit, *SIAM Journal on Scientific Computing* **20**(1): 33–61.
URL: *http://www-stat.stanford.edu/~donoho/Reports/1995/30401.pdf*

Clarke, F.H. (1990). *Optimization and Nonsmooth Analysis*, Vol. 5 of *Classics in Applied Mathemat-*

ics, SIAM, Philadelphia. Originally published by John Wiley & Sons (1983).

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2004). Least angle regression (with discussion), *Annals of Statistics* **32**(2): 407–499.

Fan, J. and Li, R. (2001). Variable selection via nonconcave penalized likelihood and its oracle properties, *Journal of the American Statistical Association* **96**(456): 1348–1360.

Fu, W.J. (1998). Penalized regression: The Bridge versus the Lasso, *Journal of Computational and Graphical Statistics* **7**(3): 397–416.

Grandvalet, Y. (1998). Least absolute shrinkage is equivalent to quadratic penalization, *in* L. Niklasson, M. Bodén and T. Ziemske (eds), *ICANN '98 , Perspectives in Neural Computing*, Vol. 1, Springer-Verlag, pp. 201–206.
URL: *http://www.hds.utc.fr/~grandval/*

Grandvalet, Y. and Canu, S. (1999). Outcomes of the equivalence of adaptive ridge with least absolute shrinkage, *in* M. Kearns, S. Solla and D. Cohn (eds), *NIPS'1998*, Vol. 11, MIT Press, pp. 445–451.
URL: *http://www.hds.utc.fr/~grandval/*

Lokhorst, J. (1999). *The LASSO and Generalised Linear Models*, Honours project, Department of Statistics, The University of Adelaide, South Australia, Australia.

McCullagh, P. and Nelder, J.A. (1989). *Generalized Linear Models*, Vol. 37 of *Monographs on Statistics and Applied Probability*, 2 edn, Chapman and Hall, London.

Osborne, M.R. (1985). *Finite Algorithms in Optimization and Data Analysis*, Wiley Series in Probability and Mathematical Statistics, John Wiley & Sons, Chichester.

Osborne, M.R., Presnell, B. and Turlach, B.A. (2000a). A new approach to variable selection in least squares problems, *IMA Journal of Numerical Analysis* **20**(3): 389–403.

Osborne, M.R., Presnell, B. and Turlach, B.A. (2000b). On the LASSO and its dual, *Journal of Computational and Graphical Statistics* **9**(2): 319–337.

Rockafellar, R.T. (1970). *Convex Analysis*, Vol. 28 of *Princeton Mathematical Series*, Princeton University Press, Princeton, New Jersey.

Rosset, S. (2004). Tracking curved regularized optimization solution paths, *NIPS'2004*, Vol. 17. to appear.

Rosset, S. and Zhu, J. (2004). Piecewise linear regularized solution paths, *unpublished manuscript*, Department of Statistics, Stanford University, Stanford, CA 94305-4065, USA.

Roth, V. (2002). The Generalized LASSO: a wrapper approach to gene selection for microarray data, *Technical Report IAI-TR-2002-8*, Institute of Computer Science III, University of Bonn, Germany.
URL: *http://www2.inf.ethz.ch/~vroth/*

Roth, V. (2004). The generalized LASSO, *IEEE Transactions on Neural Networks* **15**(1): 16–28.

Sardy, S., Bruce, A.G. and Tseng, P. (2000). Block coordinate relaxation methods for nonparametric signal denoising with wavelet dictionaries, *Journal of Computational and Graphical Statistics* **9**(2): 361–379.

Tibshirani, R. (1996). Regression Shrinkage and Selection via the Lasso, *Journal of the Royal Statistical Society, Series B* **58**(1): 267–288.