J. R. Statist. Soc. B (2005)
67, *Part* 1, *pp.* 91–108

Sparsity and smoothness via the fused lasso

Robert Tibshirani and Michael Saunders,

Stanford University, USA

Saharon Rosset,

IBM T. J. Watson Research Center, Yorktown Heights, USA

Ji Zhu

University of Michigan, Ann Arbor, USA

and Keith Knight

University of Toronto, Canada

[Received September 2003. Final revision August 2004]

Summary. The lasso penalizes a least squares regression by the sum of the absolute values (L_1-norm) of the coefficients. The form of this penalty encourages sparse solutions (with many coefficients equal to 0). We propose the 'fused lasso', a generalization that is designed for problems with features that can be ordered in some meaningful way. The fused lasso penalizes the L_1-norm of both the coefficients and their successive differences. Thus it encourages sparsity of the coefficients and also sparsity of their differences—i.e. local constancy of the coefficient profile. The fused lasso is especially useful when the number of features p is much greater than N, the sample size. The technique is also extended to the 'hinge' loss function that underlies the support vector classifier. We illustrate the methods on examples from protein mass spectroscopy and gene expression data.

Keywords: Fused lasso; Gene expression; Lasso; Least squares regression; Protein mass spectroscopy; Sparse solutions; Support vector classifier

1. Introduction

We consider a prediction problem with N cases having outcomes $y_1, y_2, \ldots, y_N$ and features x_{ij}, $i=1,2,\ldots,N$, $j=1,2,\ldots,p$. The outcome can be quantitative, or equal to 0 or 1, representing two classes like 'healthy' and 'diseased'. We also assume that the x_{ij} are realizations of features X_j that can be ordered as $X_1, X_2, \ldots, X_p$ in some meaningful way. Our goal is to predict Y from $X_1, X_2, \ldots, X_p$. We are especially interested in problems for which $p \gg N$. A motivating example comes from protein mass spectroscopy, in which we observe, for each blood serum sample i, the intensity x_{ij} for many *time-of-flight* values t_j. Time of flight is related to the mass over charge ratio m/z of the constituent proteins in the blood. Fig. 1 shows an example that is taken from Adam *et al.* (2003): the average spectra for healthy patients and those with prostate cancer. There are 48 538 m/z-sites in total. The full data set consists of 157 healthy patients and 167 with cancer, and the goal is to find m/z-sites that discriminate between the two groups.

Address for correspondence: Robert Tibshirani, Department of Health Research and Policy, H R P Redwood Building, Stanford University, Stanford, CA 94305-5405, USA.
E-mail: tibs@stat.stanford.edu

1369–7412/05/67091

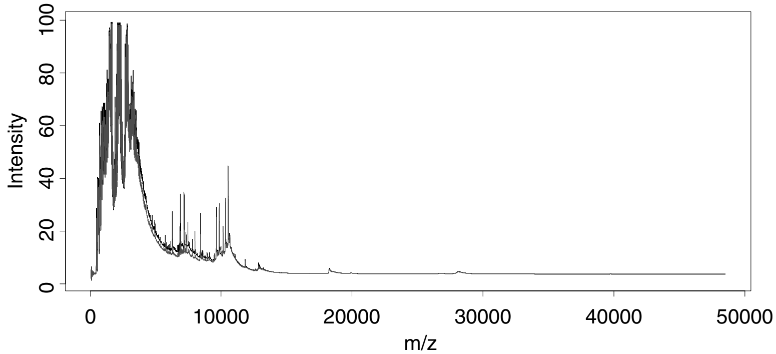


Fig. 1. Protein mass spectroscopy data: average profiles from normal (———) and prostate cancer patients (······)

There has been much interest in this problem in the past few years; see for example Petricoin *et al.* (2002) and Adam *et al.* (2003).

In other examples, the order of the features may not be fixed *a priori* but may instead be estimated from the data. An example is gene expression data measured from a microarray. Hierarchical clustering can be used to estimate an ordering of the genes, putting correlated genes near one another in the list. We illustrate our methods on both protein mass spectroscopy and microarray data in this paper.

In Section 2 we define the fused lasso and illustrate it on a simple example. Section 3 describes computation of the solutions. Section 4 explores asymptotic properties. In Section 5 we relate the fused lasso to soft threshold methods and wavelets. Degrees of freedom of the fused lasso fit are discussed in Section 6. A protein mass spectroscopy data set on prostate cancer is analysed in Section 7, whereas Section 8 carries out a simulation study. An application of the method to unordered features is discussed in Section 9 and illustrated on a microarray data set in Section 9.1. The hinge loss function and support vector classifiers are addressed in Section 10.

2. The lasso and fusion

We begin with a standard linear model

$$y_i = \sum_j x_{ij}\beta_j + \varepsilon_i \tag{1}$$

with the errors ε_i having mean 0 and constant variance. We also assume that the predictors are standardized to have mean 0 and unit variance, and the outcome y_i has mean 0. Hence we do not need an intercept in model (1).

We note that p may be larger then N, and typically it is much larger than N in the applications that we consider. Many methods have been proposed for regularized or penalized regression, including ridge regression (Hoerl and Kennard, 1970), partial least squares (Wold, 1975) and principal components regression. Subset selection is more discrete, either including or excluding predictors from the model. The lasso (Tibshirani, 1996) is similar to ridge regression but uses the absolute values of the coefficients rather than their squares. The lasso finds the coefficients $\hat{\beta} = (\hat{\beta}_1, \hat{\beta}_2, \ldots, \hat{\beta}_p)$ satisfying

$$\hat{\beta} = \arg\min\Big\{ \sum_i \Big(y_i - \sum_j x_{ij}\beta_j\Big)^2 \Big\} \qquad \text{subject to } \sum_j |\beta_j| \leqslant s. \tag{2}$$

The bound s is a tuning parameter: for sufficiently large s we obtain the least squares solution, or one of the many possible least squares solutions if $p > N$. For smaller values of s, the solutions are sparse, i.e. some components are exactly 0. This is attractive from a data analysis viewpoint, as it selects the important predictors and discards the rest. In addition, since the criterion and constraints in condition (2) are convex, the problem can be solved even for large p (e.g. $p = 40000$) by quadratic programming methods. We discuss computation in detail in Section 3.

Unlike the lasso, ridge regression, partial least squares and principal components regression do not produce sparse models. Subset selection does produce sparse models but is not a convex operation; best subsets selection is combinatorial and is not practical for $p >$ 30 or so.

The lasso can be applied even if $p > N$, and it has a unique solution assuming that no two predictors are perfectly collinear. An interesting property of the solution is the fact that the number of non-zero coefficients is at most $\min(N, p)$. Thus, if $p = 40000$ and $N = 100$, at most 100 coefficients in the solution will be non-zero. The 'basis pursuit' signal estimation method of Chen *et al.* (2001) uses the same idea as the lasso, but applied in the wavelet or other domains.

One drawback of the lasso in the present context is the fact that it ignores ordering of the features, of the type that we are assuming in this paper. For this purpose, we propose the *fused lasso* defined by

$$\hat{\beta} = \arg\min\Big\{\sum_i \Big(y_i - \sum_j x_{ij}\beta_j\Big)^2\Big\} \qquad \text{subject to } \sum_{j=1}^{p} |\beta_j| \leqslant s_1 \text{ and } \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \leqslant s_2. \tag{3}$$

The first constraint encourages sparsity in the coefficients; the second encourages sparsity in their differences, i.e. flatness of the coefficient profiles β_j as a function of j. The term fusion is borrowed from Land and Friedman (1996), who proposed the use of a penalty of the form $\Sigma_j |\beta_j - \beta_{j-1}|^\alpha \leqslant s_2$ for various values of α, especially $\alpha = 0, 1, 2$. They did not consider the use of penalties on both $\Sigma_j |\beta_j - \beta_{j-1}|$ *and* $\Sigma_j |\beta_j|$ as in condition (3). Fig. 2 gives a schematic view.

Fig. 3 illustrates these ideas on a simulated example. There are $p = 100$ predictors and $N = 20$ samples. The data were generated from a model $y_i = \Sigma_j x_{ij}\beta_j + \varepsilon_i$ where the x_{ij} are standard

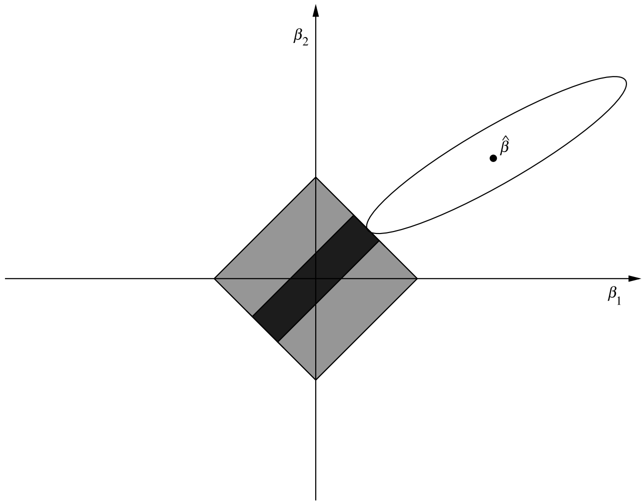


Fig. 2. Schematic diagram of the fused lasso, for the case $N > p = 2$: we seek the first time that the contours of the sum-of-squares loss function (⬭) satisfy $\Sigma_j |\beta_j| = s_1$ (◆) and $\Sigma_j |\beta_j - \beta_{j-1}| = s_2$ (⬛)

Gaussian, $\varepsilon_i \sim N(0,\sigma^2)$ with $\sigma=0.75$, and there are three blocks of consecutive non-zero β_js shown by the black points in each of the panels. Fig. 3(a) shows the univariate regression coefficients (red) and a soft thresholded version of them (green). Fig. 3(b) shows the lasso solution (red), using $s_1=35.6$ and $s_2=\infty$, and Fig. 3(c) shows the fusion estimate (using $s_1=\infty$ and $s_2=26$). These values of s_1 and s_2 were the values that minimized the estimated test set error. Finally Fig. 3(d) shows the fused lasso, using $s_1=\Sigma_j\,|\beta_j|$ and $s_2=\Sigma_j\,|\beta_j-\beta_{j-1}|$, where β is the true set of coefficients. The fused lasso does the best job in estimating the true underlying coefficients. However, the fusion method (Fig. 3(c)) performs as well as the fused lasso does in this example.

Fig. 4 shows another example, with the same set-up as in Fig. 3 except that $\sigma=0.05$ and β has two non-zero areas—a spike at $m/z=10$ and a flat plateau between 70 and 90. As in the previous example, the bounds s_1 and s_2 were chosen in each case to minimize the prediction error. The lasso performs poorly; fusion captures the plateau but does not clearly isolate the peak at $m/z=10$. The fused lasso does a good job overall.

An alternative formulation would use a second penalty of the form $\Sigma_j(\beta_j-\beta_{j-1})^2 \leqslant s_2$ in place of $\Sigma_j|\beta_j-\beta_{j-1}| \leqslant s_2$ (which was also suggested by a referee). However, this has the analogous drawback that $\Sigma\,\beta_j^2$ has compared with $\Sigma_j|\beta_j|$: it does not produce a sparse solution, where sparsity refers to the first differences $\beta_j-\beta_{j-1}$. The penalty $\Sigma_j\,(\beta_j-\beta_{j-1})^2 \leqslant s_2$ does not produce a simple piecewise constant solution, but rather a 'wiggly' solution that is less attractive for interpretation. The penalty $\Sigma_j|\beta_j-\beta_{j-1}| \leqslant s_2$ gives a piecewise constant solution, and this corresponds to a simple averaging of the features.

3. Computational approach

3.1. Fixed s_1 and s_2

Criterion (3) leads to a quadratic programming problem. For large p, the problem is difficult to solve and special care must be taken to avoid the use of p^2 storage elements. We use the two-phase active set algorithm SQOPT of Gill *et al.* (1997), which is designed for quadratic programming problems with sparse linear constraints.

Let $\beta_j=\beta_j^+-\beta_j^-$ with $\beta_j^+,\beta_j^- \geqslant 0$. Define $\theta_j=\beta_j-\beta_{j-1}$ for $j>1$ and $\theta_1=\beta_1$. Let $\theta_j=\theta_j^+-\theta_j^-$ with $\theta_j^+,\theta_j^- \geqslant 0$. Let L be a $p\times p$ matrix with $L_{ii}=1$, $L_{i+1,i}=-1$ and $L_{ij}=0$ otherwise so that $\theta=L\beta$. Let e be a column p-vector of 1s, and I be the $p\times p$ identity matrix.

Let X be the $N\times p$ matrix of features and y and β be N- and p-vectors of outcomes and coefficients respectively. We can write problem (3) as

$$\hat{\beta}=\arg\min\{(y-X\beta)^{\mathrm{T}}s(y-X\beta)\} \tag{4}$$

subject to

$$\begin{pmatrix}-a_0\\0\\0\\0\end{pmatrix} \leqslant \begin{pmatrix}L & 0 & 0 & -I & I\\ I & -I & I & 0 & 0\\ 0 & e^{\mathrm{T}} & e^{\mathrm{T}} & 0 & 0\\ 0 & 0 & 0 & e_0^{\mathrm{T}} & e_0^{\mathrm{T}}\end{pmatrix}\begin{pmatrix}\beta\\ \beta^+\\ \beta^-\\ \theta^+\\ \theta^-\end{pmatrix} \leqslant \begin{pmatrix}a_0\\0\\s_1\\s_2\end{pmatrix}, \tag{5}$$

in addition to the non-negativity constraints $\beta^+,\beta^-,\theta^+,\theta^- \geqslant 0$. The big matrix is of dimension $(2p+2)\times 5p$ but has only $11p-1$ non-zero elements. Here $a_0=(\infty,0,0,\ldots,0)$. Since $\beta_1=\theta_1$, setting its bounds at $\pm\infty$ avoids a double penalty for $|\beta_1|$. Similarly $e_0=e$ with the first component set to 0.

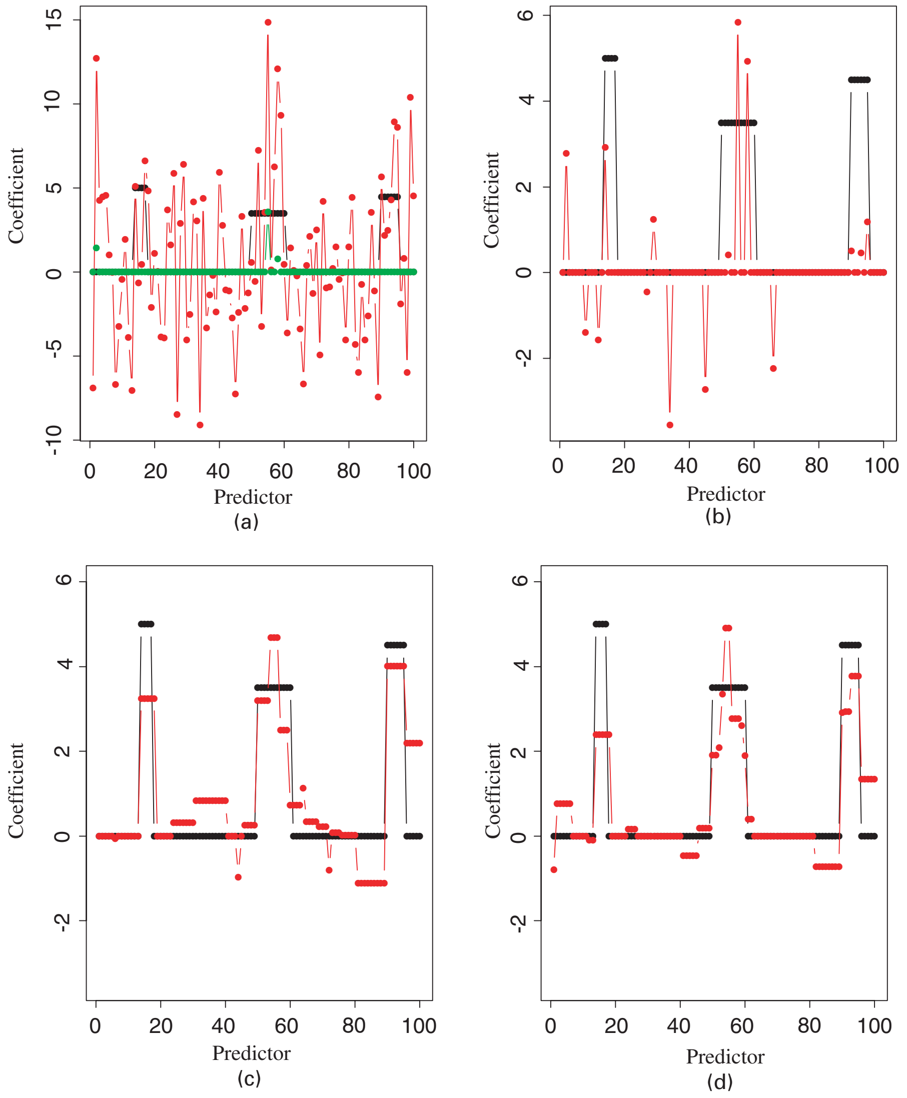


Fig. 3. Simulated example, with $p=100$ predictors having coefficients shown by the black lines: (a) univariate regression coefficients (red) and a soft thresholded version of them (green); (b) lasso solution (red), using $s_1=35.6$ and $s_2=\infty$; (c) fusion estimate, using $s_1=\infty$ and $s_2=26$ (these values of s_1 and s_2 minimized the estimated test set error); (d) the fused lasso, using $s_1=\Sigma_j|\beta_j|$ and $s_2=\Sigma_j|\beta_j-\beta_{j-1}|$, where β is the true set of coefficients

The SQOPT package requires the user to write a procedure that computes $X^{\mathrm{T}}Xv$ for p-vectors v that are under consideration. For many choices of the bounds s_1 and s_2, the vector v is very sparse and hence $X^{\mathrm{T}}(Xv)$ can be efficiently computed. The algorithm is also well suited for ‘warm starts’: starting at a solution for a given s_1 and s_2, the solution for nearby values of these bounds can be found relatively quickly.

3.2. *Search strategy*

For moderate-sized problems ($p\simeq 1000$ and $N\simeq 100$ say), the above procedure is sufficiently

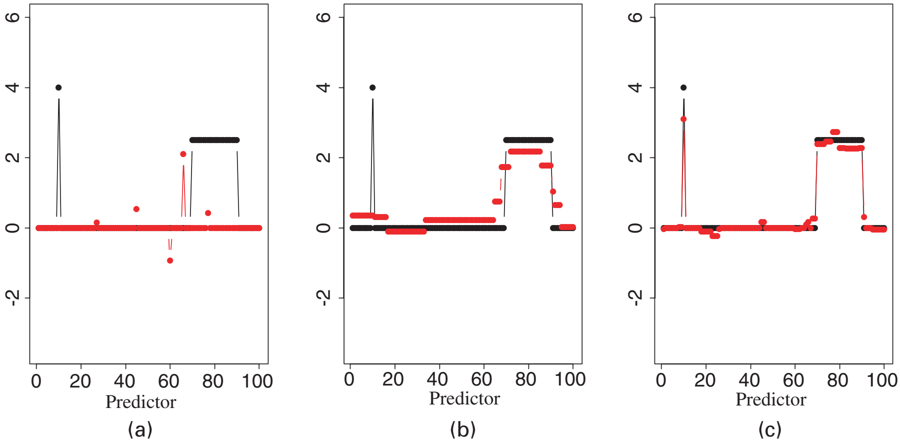


Fig. 4. Simulated example with only two areas of non-zero coefficients (black points and lines; red points, estimated coefficients from each method): (a) lasso, $s_1 = 4.2$; (b) fusion, $s_2 = 5.2$; (c) fused lasso, $s_1 = 56.5$, $s_2 = 13$

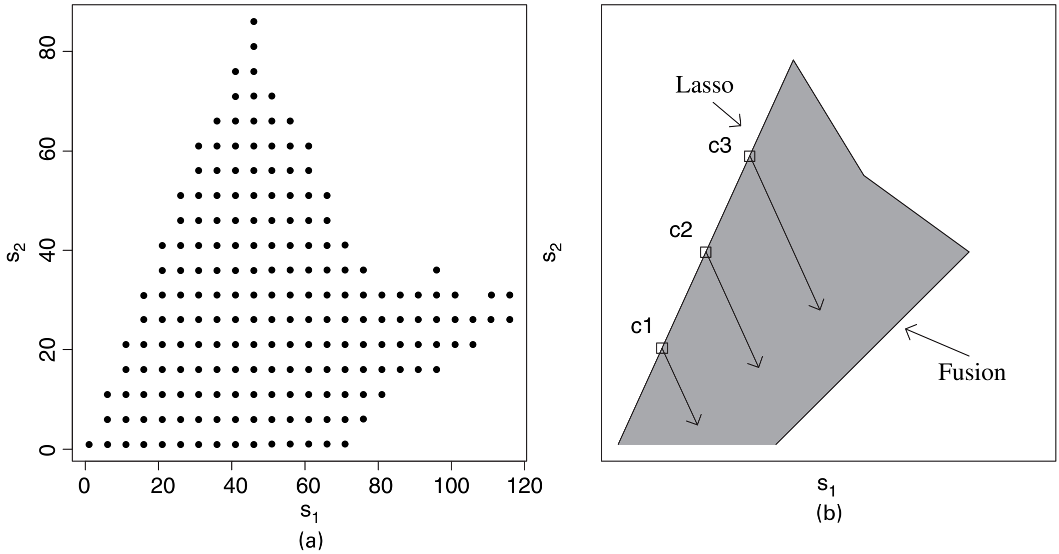


Fig. 5. Simulated example of Fig. 3: (a) attainable values of bounds s_1 and s_2; (b) schematic diagram of the search process for the fused lasso, described in the text

fast that it can be applied over a grid of s_1- and s_2-values. For larger problems, a more restricted search is necessary. We first exploit the fact that the complete sequence of lasso and fusion problems can be solved efficiently by using the least angle regression (LAR) procedure (Efron *et al.*, 2002). The fusion problem is solved by first transforming X to $Z = XL^{-1}$ with $\theta = L\beta$, applying LAR and then transforming back.

For a given problem, only some values of the bounds (s_1, s_2) will be attainable, i.e. the solution vector satisfies both $\Sigma_j |\hat{\beta}_j| = s_1$ and $\Sigma_j |\hat{\beta}_j - \hat{\beta}_{j-1}| = s_2$. Fig. 5(a) shows the attainable values for our simulated data example.

Fig. 5(b) is a schematic diagram of the search strategy. Using the LAR procedure as above, we obtain solutions for bounds $(s_1(i), \infty)$, where $s_1(i)$ is the bound giving a solution with i degrees

Table 1. Timings for typical runs of the fused lasso program

p	*N*	*Start*	*Time (s)*
100	20	Cold	0.09
500	20	Cold	1.0
1000	20	Cold	2.0
1000	200	Cold	30.4
2000	200	Cold	120.0
2000	200	Warm	16.6

of freedom. (We discuss the 'degrees of freedom' of the fused lasso fit in Section 6.) We use the lasso sequence of solutions and cross-validation or a test set to estimate an optimal degrees of freedom $\hat{\imath}$. Now let

$$s_{2\max}\{s_1(\hat{\imath})\}=\sum_j |\hat{\beta}_j\{s_1(\hat{\imath})\}-\hat{\beta}_{j-1}\{s_1(\hat{\imath})\}|.$$

This is the largest value of the bound s_2 at which it affects the solution. The point c_2 in Fig. 5(b) is $[s_1(\hat{\imath}), s_{2\max}\{s_1(\hat{\imath})\}]$. We start at c_2 and fuse the solutions by moving in the direction $(1, -2)$. In the same way, we define points c_1 to be the solution with degrees of freedom $\hat{\imath}/2$ and c_3 to have degrees of freedom $\{\hat{\imath}+\min(N, p)\}/2$, and we fuse the solutions from those points. The particular direction $(1, -2)$ was chosen by empirical experimentation. We are typically not interested in solutions that are near the pure fusion model (the lower right boundary), and this search strategy tries to cover (roughly) the potentially useful values of (s_1, s_2). This strategy is used in the real examples and simulation study that are discussed later in the paper.

For real data sets, we apply this search strategy to a training set and then evaluate the prediction error over a validation set. This can be done with a single training–validation split, or through fivefold or tenfold cross-validation. These are illustrated in the examples later in the paper.

Table 1 shows some typical computation times for problems of various dimensions, on a 2.4 GHz Xeon Linux computer. Some further discussion of computational issues can be found in Section 11.

4. Asymptotic properties

In this section we derive results for the fused lasso that are analogous to those for the lasso (Knight and Fu, 2000). The penalized least squares criterion is

$$\sum_{i=1}^{N}(y_i-\mathbf{x}_i^{\mathrm{T}}\beta)^2+\lambda_N^{(1)}\sum_{j=1}^{p}|\beta_j|+\lambda_N^{(2)}\sum_{j=2}^{p}|\beta_j-\beta_{j-1}| \tag{6}$$

with $\beta=(\beta_1,\beta_2,\ldots,\beta_p)^{\mathrm{T}}$ and $\mathbf{x}_i=(x_{i1},x_{i2},\ldots x_{ip})^{\mathrm{T}}$, and the Lagrange multipliers $\lambda_N^{(1)}$ and $\lambda_N^{(2)}$ are functions of the sample size N.

For simplicity, we assume that p is fixed with $N\rightarrow\infty$. These are not particularly realistic asymptotic conditions: we would prefer to have $p=p_N\rightarrow\infty$ as $N\rightarrow\infty$. A result along these lines is probably attainable. However, the following theorem adequately illustrates the basic dynamics of the fused lasso.

Theorem 1. If $\lambda_N^{(l)}/\surd N \rightarrow \lambda_0^{(l)} \geqslant 0$ $(l=1,2)$ and

$$C=\lim_{N\rightarrow\infty}\left(\frac{1}{N}\sum_{i=1}^{N}\mathbf{x}_i\mathbf{x}_i^{\mathrm{T}}\right)$$

is non-singular then

$$\surd N(\hat{\beta}_N-\beta)\xrightarrow[\mathrm{d}]{}\arg\min(V),$$

where

$$V(\mathbf{u})=-2\mathbf{u}^{\mathrm{T}}\mathbf{W}+\mathbf{u}^{\mathrm{T}}C\mathbf{u}+\lambda_0^{(1)}\sum_{j=1}^{p}\{u_j\,\mathrm{sgn}(\beta_j)\,I(\beta_j\neq 0)+|u_j|\,I(\beta_j=0)\}$$
$$+\lambda_0^{(2)}\sum_{j=2}^{p}\{(u_j-u_{j-1})\,\mathrm{sgn}(\beta_j-\beta_{j-1})\,I(\beta_j\neq\beta_{j-1})+|u_j-u_{j-1}|\,I(\beta_j=\beta_{j-1})\}$$

and $\mathbf{W}$ has an $\mathcal{N}(\mathbf{0},\sigma^2C)$ distribution.

Proof. Define $V_N(\mathbf{u})$ by

$$V_N(\mathbf{u})=\sum_{i=1}^{N}\{(\varepsilon_i-\mathbf{u}^{\mathrm{T}}\mathbf{x}_i/\surd N)^2-\varepsilon_i^2\}+\lambda_N^{(1)}\sum_{j=1}^{p}(|\beta_j+u_j/\surd N|-|\beta_j|)$$
$$+\lambda_N^{(2)}\sum_{j=2}^{p}\{|\beta_j-\beta_{j-1}+(u_j-u_{j-1})/\surd N|-|\beta_j-\beta_{j-1}|\}$$

with $\mathbf{u}=(u_0,u_1,\ldots,u_p)^{\mathrm{T}}$, and note that V_N is minimized at $\surd N(\hat{\beta}_N-\beta)$. First note that

$$\sum_{i=1}^{N}\{(\varepsilon_i-\mathbf{u}^{\mathrm{T}}\mathbf{x}_i/\surd N)^2-\varepsilon_i^2\}\xrightarrow[\mathrm{d}]{}-2\mathbf{u}^{\mathrm{T}}\mathbf{W}+\mathbf{u}^{\mathrm{T}}C\mathbf{u}$$

with finite dimensional convergence holding trivially. We also have

$$\lambda_N^{(1)}\sum_{j=1}^{p}(|\beta_j+u_j/\surd N|-|\beta_j|)\rightarrow\lambda_0^{(1)}\sum_{j=1}^{p}\{u_j\,\mathrm{sgn}(\beta_j)\,I(\beta_j\neq 0)+|u_j|\,I(\beta_j=0)\}$$

and

$$\lambda_N^{(2)}\sum_{j=2}^{p}\{|\beta_j-\beta_{j-1}+(u_j-u_{j-1})/\surd N|-|\beta_j-\beta_{j-1}|\}\rightarrow$$
$$\lambda_0^{(2)}\sum_{j=2}^{p}\{(u_j-u_{j-1})\,\mathrm{sgn}(\beta_j-\beta_{j-1})\,I(\beta_j\neq\beta_{j-1})\}+\lambda_0^{(2)}\sum_{j=2}^{p}\{|u_j-u_{j-1}|\,I(\beta_j=\beta_{j-1})\}.$$

Thus $V_N(\mathbf{u})\rightarrow_{\mathrm{d}} V(\mathbf{u})$ (as defined above), with finite dimensional convergence holding trivially. Since V_N is convex and V has a unique minimum, it follows (Geyer, 1996) that

$$\arg\min(V_N)=\surd N(\hat{\beta}_N-\beta)\xrightarrow[\mathrm{d}]{}\arg\min(V). \qquad \square$$

As a simple example, suppose that $\beta_1=\beta_2\neq 0$. Then the joint limiting distribution of

$$(\surd N(\hat{\beta}_{1N}-\beta_1),\surd N(\hat{\beta}_{2N}-\beta_2))$$

will have probability concentrated on the line $u_1=u_2$ when $\lambda_0^{(2)}>0$. When $\lambda_0^{(1)}>0$, we would see a lasso-type effect on the univariate limiting distributions, which would result in a shift of probability to the negative side if $\beta_1 = \beta_2 > 0$ and a shift of probability to the positive side if $\beta_1 = \beta_2 < 0$.

5. Soft thresholding and wavelets

5.1. Soft thresholding estimators

Consider first the lasso problem with orthonormal features and $N > p$, i.e. in the fused lasso problem (3) we take $s_2 = \infty$ and we assume that $X^{\mathrm{T}}X = I$. Then, if $\tilde{\beta}_j$ are the univariate least squares estimates, the lasso solutions are soft threshold estimates:

$$\hat{\beta}_j(\gamma_1) = \operatorname{sgn}(\tilde{\beta}_j) \cdot (|\tilde{\beta}_j| - \gamma_1)_+, \tag{7}$$

where γ_1 satisfies $\Sigma_j |\hat{\beta}_j(\gamma_1)| = s_1$.

Corresponding to this, there is a special case of the fused problem that also has an explicit solution. We take $s_1 = \infty$ and let $\theta = L\beta$ and $Z = XL^{-1}$. Note that L^{-1} is a lower triangular matrix of 1s, and hence the components of Z are the 'right' cumulative sums of the x_{ij} across j. This gives a lasso problem for (Z, y) and the solutions are

$$\hat{\theta}_j(\gamma_2) = \operatorname{sgn}(\tilde{\theta}_j) \cdot (|\tilde{\theta}_j| - \gamma_2)_+, \tag{8}$$

provided that $Z^{\mathrm{T}}Z = I$, or equivalently $X^{\mathrm{T}}X = L^{\mathrm{T}}L$. Here γ_2 satisfies $\Sigma_j |\hat{\theta}_j(\gamma_2)| = s_2$. The matrix $L^{\mathrm{T}}L$ is tridiagonal, with 2s on the diagonal and -1s on the off-diagonals.

Of course we cannot have both $X^{\mathrm{T}}X = I$ and $X^{\mathrm{T}}X = L^{\mathrm{T}}L$ at the same time. But we can construct a scenario for which the fused lasso problem has an explicit solution. We take $X = UL^{-1}$ with $U^{\mathrm{T}}U = I$ and assume that the full least squares estimates $\beta' = (X^{\mathrm{T}}X)^{-1}X^{\mathrm{T}}y$ are non-decreasing: $0 \leqslant \beta'_1 \leqslant \beta'_2 \leqslant \ldots \leqslant \beta'_p$. Finally, we set $s_1 = s_2 = s$. Then the fused lasso solution soft-thresholds the full least squares estimates β' from the right:

$$\hat{\beta} = (\beta'_1, \beta'_2, \ldots \beta'_k, \lambda, 0, 0, \ldots 0), \tag{9}$$

where $\Sigma_1^k \beta'_j + \lambda = s$. However, this set-up does not seem to be very useful in practice, as its assumptions are quite unrealistic.

5.2. Basis transformations

A transform approach to the problem of this paper would go roughly as follows. We model $\beta = W\gamma$, where the columns of W are appropriate bases. For example, in our simulated example we might use Haar wavelets, and then we can write $X\beta = X(W\gamma) = (XW)\gamma$. Operationally, we transform our features to $Z = XW$ and fit y to $Z\gamma$, either by soft thresholding or by lasso, giving $\tilde{\gamma}$. Finally we map back to obtain $\tilde{\beta} = W\tilde{\gamma}$. Note that soft thresholding implicitly assumes that the Z-basis is orthonormal: $Z^{\mathrm{T}}Z = I$.

This procedure seeks a sparse representation of the βs in the transformed space. In contrast, the lasso and simple soft thresholded estimates (7) seek a sparse representation of the βs in the original basis.

The fused lasso is more ambitious: it uses two basis representations X and $Z = XL^{-1}$ and seeks a representation that is sparse in both spaces. It does not assume orthogonality, since this cannot hold simultaneously in both representations. The price for this ambition is an increased computational burden.

Fig. 6 shows the results of applying soft thresholding (Fig. 6(a)) or the lasso (Fig. 6(b)) in the space of Haar wavelets coefficients, and then transforming back to the original space. For soft

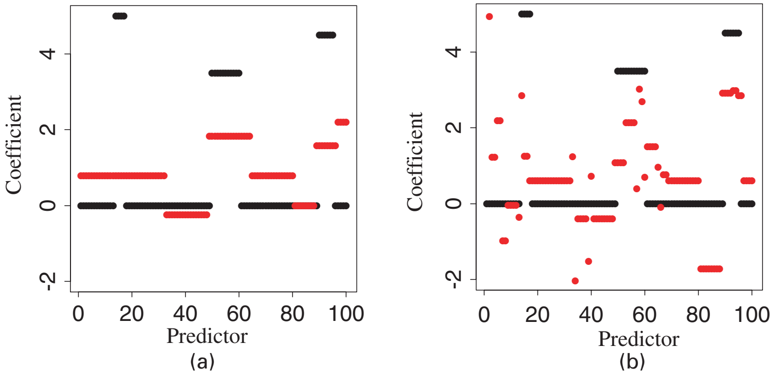


Fig. 6. Simulated example of Fig. 3: (a) true coefficients (black), and estimated coefficients (red) obtained from transforming to a Haar wavelet basis, thresholding and transforming back; (b) same procedure, except that the lasso was applied to the Haar coefficients (rather than soft thresholding)

thresholding, we used the level-dependent threshold $\sigma\surd\{2\log(N_j)\}$, where N_j is the number of wavelet coefficients at the given scale and σ was chosen to minimize the test error (see for example Donoho and Johnstone (1994)). For the lasso, we chose the bound s_1 to minimize the test error. The resulting estimates are not very accurate, especially that from the lasso. This may be partly due to the fact that the wavelet basis is not translation invariant. Hence, if the non-zero coefficients are not situated near a power of 2 along the feature axis, the wavelet basis will have difficulty representing it.

6. Degrees of freedom of the fused lasso fit

It is useful to consider how many 'degrees of freedom' are used in a fused lasso fit $\hat{y}=X\hat{\beta}$ as s_1 and s_2 are varied. Efron *et al.* (2002) considered a definition of degrees of freedom using the formula of Stein (1981):

$$\mathrm{df}(\hat{y})=\frac{1}{\sigma^2}\sum_{i=1}^{N}\mathrm{cov}(y_i,\hat{y}_i), \tag{10}$$

where σ^2 is the variance of y_i with X fixed and cov refers to covariance with X fixed. For a standard multiple linear regression with $p<N$ predictors, $\mathrm{df}(\hat{y})$ reduces to p. Now, in the special case of an orthonormal design ($X^{\mathrm{T}}X=I$), the lasso estimators are simply the soft threshold estimates (7), and Efron *et al.* (2002) showed that the degrees of freedom equal the number of non-zero coefficients. They also proved this for the LAR and lasso estimators under a 'positive cone condition', which implies that the estimates are monotone as a function of the L_1-bound s_1. The proof in the orthonormal case is simple: it uses Stein's formula

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}\mathrm{cov}(y_i,g_i)=E\left\{\sum_i\frac{\partial g(y)}{\partial y_i}\right\}, \tag{11}$$

where $y=(y_1,y_2,\ldots,y_N)$ is a multivariate normal vector with mean μ and covariance I, and $g(y)$ is an estimator, an almost differentiable function from $\mathbb{R}^N$ to $\mathbb{R}^N$. For the lasso with orthonormal design, we rotate the basis so that $X=I$, and hence from equation (7) $g(y)=\mathrm{sgn}(y_i)(|y_i|-\gamma_1)$. The derivative $\partial g(y)/\partial y_i$ equals 1 if the ith component is non-zero and 0 otherwise. Hence the degrees of freedom are the number of non-zero coefficients.

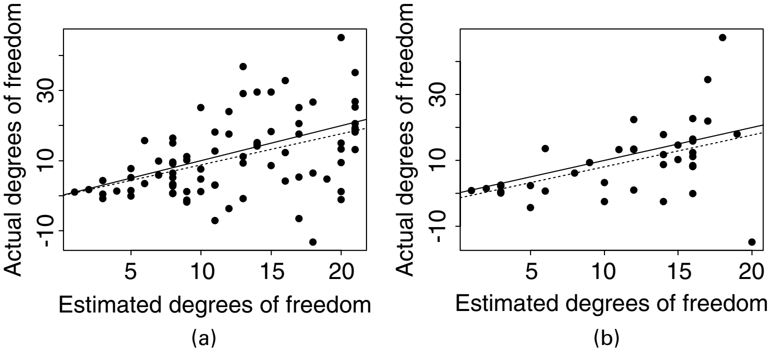


Fig. 7. Simulated example: actual and estimated degrees of freedom for (a) the fused lasso and (b) the lasso (———, 45°-line; - - - - - -, least squares regression fit)

For the fused lasso, the natural estimate of the degrees of freedom is

$$\mathrm{df}(\hat{y}) = \#\{\text{non-zero coefficient blocks in } \hat{\beta}\}. \tag{12}$$

In other words, we count a sequence of one or more consecutive non-zero and equal $\hat{\beta}_j$-values as 1 degree of freedom. Equivalently, we can define

$$\mathrm{df}(\hat{y}) = p - \#\{\beta_j = 0\} - \#\{\beta_j - \beta_{j-1} = 0,\ \beta_j, \beta_{j-1} \neq 0\}. \tag{13}$$

It is easy to see that these two definitions are the same. Furthermore, the objective function can be made 0 when $\mathrm{df}(\hat{y}) \geqslant \min(N, p)$, and hence $\min(N, p)$ is an effective upper bound for the degrees of freedom. We have no proof that $\mathrm{df}(\hat{y})$ is a good estimate in general, but it follows from the Stein result (11) in scenarios (7)–(9).

Fig. 7 compares the estimated and actual degrees of freedom for the fused lasso and the lasso. The approximation for the fused lasso is fairly crude, but it is not much worse than that for the lasso. We used this definition only for descriptive purposes, to obtain a rough idea of the complexity of the fitted model.

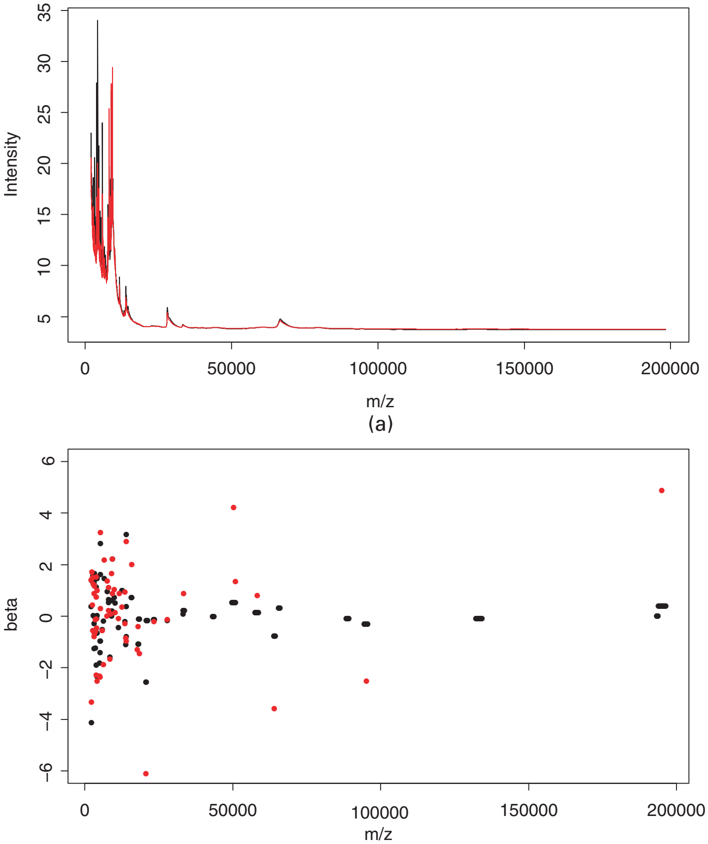
6.1. Sparsity of fused lasso solutions

As was mentioned in Section 2, the lasso has a sparse solution in high dimensional modelling, i.e., if $p > N$, lasso solutions will have at most N non-zero coefficients, under mild ('non-redundancy') conditions. This property extends to any convex loss function with a lasso penalty. It is proven explicitly, and the required non-redundancy conditions are spelled out, in Rosset *et al.* (2004), appendix A.

The fused lasso turns out to have a similar sparsity property. Instead of applying to the number of non-zero coefficients, however, the sparsity property applies to the number of sequences of identical non-zero coefficients. So, if we consider the prostate cancer example in Section 7 and Fig. 8, sparsity of the lasso implies that we could have at most 216 red dots in Fig. 8(b). Sparsity of the fused lasso implies that we could have at most 216 black sequences of consecutive m/z-values with the same coefficient.

The formal statement of the sparsity result for the fused lasso follows.

Theorem 2. Set $\beta_0 = 0$. Let $n_{\text{seq}}(\beta) = \Sigma_{j=1}^{p} \mathbf{1}\{\beta_j \neq \beta_{j-1}\}$. Then, under 'non-redundancy' conditions on the design matrix X, the fused lasso problem (3) has a unique solution $\hat{\beta}$ with $n_{\text{seq}}(\hat{\beta}) \leqslant N$.



(a)

(b)

Fig. 8. Results for the prostate cancer example: ———, ●, fused lasso non-zero coefficients; ··········, ●, lasso non-zero coefficients

The proof is very similar to the sparsity proof for the lasso in Rosset *et al.* (2004), and is based on examining the Karush–Kuhn–Tucker conditions for optimality of the solution to the constrained problem (3). The non-redundancy conditions mentioned can be qualitatively summarized as follows.

(a) No N columns of the design matrix X are linearly dependent.
(b) None of a finite set of $N+1$ linear equations in N variables (the coefficients of which depend on the specific problem) has a solution.

7. Analysis of prostate cancer data

As mentioned in Section 1 the prostate cancer data set consists of 48 538 measurements on 324 patients: 157 healthy and 167 with cancer. The average profiles (centroids) are shown in Fig. 1. Following the original researchers, we ignored m/z-sites below 2000, where chemical artefacts can occur. We randomly created training and validation sets of size 216 and 108 patients respectively. To make computations manageable, we average the data in consecutive blocks of 20, giving a total of 2181 sites. (We did manage to run the lasso on the full set of sites, and it produced error rates that were about the same as those reported for the lasso here.) The

Table 2. Prostate data results

Method	*Validation errors/108*	*Degrees of freedom*	*Number of sites*	s_1	s_2
Nearest shrunken centroids	30		227		
Lasso	16	60	40	83	164
Fusion	18	102	2171	16	32
Fused lasso	16	103	218	113	103

results of various methods are shown in Table 2. In this two-class setting, the 'nearest shrunken centroids' method (Tibshirani *et al.*, 2001) is essentially equivalent to soft thresholding of the univariate regression coefficients.

Adam *et al.* (2003) reported error rates around 5% for a four-class version of this problem, using a peak finding procedure followed by a decision tree algorithm. However, we (and at least one other group that we know of) have had difficulty replicating their results, even when using their extracted peaks.

Fig. 8 shows the non-zero coefficients from the two methods. We see that the fused lasso puts non-zero weights at more sites, spreading out the weights especially at higher m/z-values. A more careful analysis would use cross-validation to choose the bounds, and then report the test error for these bounds. We carry out such an analysis for the leukaemia data in Section 9.1.

8. A simulation study

We carried out a small simulation study to compare the performance of the lasso and the fused lasso. To ensure that our feature set had a realistic correlation structure for protein mass spectroscopy, we used the first 1000 features from the data set that was described in the previous section. We also used a random subset of 100 of the patients, to keep the feature to sample size ratio near a realistic level. We then generated coefficient vectors β by choosing 1–10 non-overlapping m/z-sites at random and defining blocks of equal non-zero coefficients of lengths uniform between 1 and 100. The values of the coefficients were generated as $N(0,1)$. Finally, training and test sets were generated according to

$$\begin{aligned} y &= X\beta + Z, \\ 2.5Z &\sim N(0,1). \end{aligned} \tag{14}$$

The set-up is such that the amount of test variance that is explained by the model is about 50%.

For each data set, we found the lasso solution with the minimum test error. We then used the search strategy that was outlined in Section 3 for the fused lasso. Table 3 summarizes the results of 20 simulations from this model. Sensitivity and specificity refer to the proportion of true non-zero coefficients and true zero coefficients that are detected by each method. Shown are the minimum test error solution from the fused lasso and also that for the true values of the bounds s_1 and s_2.

We see that the fused lasso slightly improves on the test error of the lasso and detects a much large proportion of the true non-zero coefficients. In the process, it has a lower specificity. Even with the true s_1- and s_2-bounds, the fused lasso detects less than half the true non-zero coefficients. This demonstrates the inherent difficulty of problems having $p \gg N$.

Table 3. Results of the simulation study†

Method	*Test error*	*Sensitivity*	*Specificity*
Lasso	265.194 (7.957)	0.055 (0.009)	0.985 (0.003)
Fused lasso	256.117 (7.450)	0.478 (0.082)	0.693 (0.072)
Fused lasso (true s_1, s_2)	261.380 (8.724)	0.446 (0.045)	0.832 (0.018)

†Standard errors are given in parentheses.

9. Application to unordered features

The fused lasso definition (3) assumes that the features x_{ij}, and hence the corresponding parameters β_j, have a natural order in j. In some problems, however, the features have no prespecified order, e.g. genes in a microarray experiment. There are at least two ways to apply the fused lasso in this case. First, we can estimate an order for the features, using for example multidimensional scaling or hierarchical clustering. The latter is commonly used for creating heat map displays of microarray data.

Alternatively, we notice that definition (3) does not require a complete ordering of the features but only specification of the nearest neighbour of each feature, i.e. let $k(j)$ be the index of the feature that is closest to feature j, in terms, for example, of the smallest Euclidean distance or maximal correlation. Then we can use the fused lasso with difference constraint

$$\sum_j |\beta_j - \beta_{k(j)}| \leqslant s_2.$$

Computationally, this just changes the p linear constraints that are expressed in matrix L in expression (5). Note that more complicated schemes, such as the use of more than one near neighbour, would increase the number of linear constraints, potentially up to p^2. We illustrate the first method in the example below.

9.1. Leukaemia classification by using microarrays

The leukaemia data were introduced in Golub *et al.* (1999). There are 7129 genes and 38 samples: 27 in class 1 (acute lymphocytic leukaemia) and 11 in class 2 (acute mylogenous leukaemia). In addition there is a test sample of size 34. The prediction results are shown in Table 4.

The first two rows are based on all 7129 genes. The procedure of Golub *et al.* (1999) is similar to nearest shrunken centroids, but it uses hard thresholding. For the lasso and fusion methods, we first filtered down to the top 1000 genes in terms of overall variance. Then we applied average linkage hierarchical clustering to the genes, to provide a gene order for the fusion process.

All lasso and fusion models were fitted by optimizing the tuning parameters using cross-validation and then applying these values to the test set. The pure fusion estimate method (6) did poorly in the test error: this error never dropped below 3 for any value of the bound s_2.

We see that in row (4) fusing the lasso solution gives about the same error rate, using about four times as many genes. Further fusion in row (5) seems to increase the test error rate. Table 5 shows a sample of the estimated coefficients for the lasso and fused lasso solution method (4). We see that in many cases the fusion process has spread out the coefficient of a non-zero lasso coefficient onto adjacent genes.

Table 4. Results for the leukaemia microarray example

Method	*10-fold cross-validation error*	*Test error*	*Number of genes*
(1) Golub *et al.* (1999) (50 genes)	3/38	4/34	50
(2) Nearest shrunken centroid (21 genes)	1/38	2/34	21
(3) Lasso, 37 degrees of freedom ($s_1 = 0.65, s_2 = 1.32$)	1/38	1/34	37
(4) Fused lasso, 38 degrees of freedom ($s_1 = 1.08, s_2 = 0.71$)	1/38	2/34	135
(5) Fused lasso, 20 degrees of freedom ($s_1 = 1.35, s_2 = 1.01$)	1/38	4/34	737
(6) Fusion, 1 degree of freedom	1/38	12/34	975

Table 5. Leukaemia data example: a sample of the non-zero coefficients for the lasso and fused lasso, with contiguous blocks delineated†

Gene	*Lasso*	*Fused lasso*	*Gene*	*Lasso*	*Fused lasso*	*Gene*	*Lasso*	*Fused lasso*
9	0.00000	0.00203	421	−0.08874	−0.02506	765	0.00000	0.00361
10	0.00000	0.00495	422	0.00000	−0.00110	766	0.00000	0.00361
11	0.00000	0.00495				767	0.00000	0.00361
12	0.00000	0.00495	475	−0.01734	0.00000	768	0.00000	0.00361
13	0.00000	0.00495				769	0.00102	0.00361
14	0.00000	0.00495	522	0.00000	−0.00907	770	0.00000	0.00361
15	0.00000	0.00495	523	0.00000	−0.00907	771	0.00000	0.00361
			524	0.00000	−0.00907	772	0.00000	0.00361
22	0.01923	0.00745	525	0.00000	−0.00907			
23	0.00000	0.00745	526	0.00000	−0.00907	788	0.04317	0.03327
24	0.00000	0.00745	527	0.00000	−0.00907			
25	0.00000	0.00745	528	0.00000	−0.00907	798	0.02476	0.01514
26	0.00000	0.00745				799	0.00000	0.01514
27	0.01157	0.00294	530	0.01062	0.00000	800	0.00000	0.01514
31	−0.00227	0.00000	563	0.00000	−0.02018	815	−0.00239	0.00000
			564	0.00000	−0.02018			
39	−0.00992	0.00000	565	0.00000	−0.02018	835	0.00000	−0.01996
			566	0.00000	−0.02018	836	0.00000	−0.01996
44	−0.00181	0.00000	567	0.00000	−0.02018	837	0.00000	−0.01996
						838	0.00000	−0.00408

†The full table appears in Tibshirani *et al.* (2004).

10. Hinge loss

For two-class problems the maximum margin approach that is used in the support vector classifier (Boser *et al.*, 1992; Vapnik, 1996) is an attractive alternative to least squares. The maximum margin method can be expressed in terms of the 'hinge' loss function (see for example Hastie *et al.* (2001), chapter 11). We minimize

$$J(\beta_0, \beta, \xi) = \sum_{i=1}^{N} \xi_i \tag{15}$$

Table 6. Signs of fused lasso coefficients (rows) *versus* signs of fused lasso support vector coefficients (columns)

	–1	0	1
–1	12	28	0
0	17	822	26
1	0	60	35

subject to

$$y_i(\beta_0+\beta^{\mathrm{T}}\mathbf{x}_i)\geqslant 1-\xi_i, \qquad \xi_i\geqslant 0, \text{ for all } i.$$

The original support vector classifier includes an L_2-constraint $\Sigma_{j=1}^{p}\beta_j^2\leqslant s$. Recently there has been interest in the L_1-constrained (lasso) support vector classifier. Zhu *et al.* (2003) developed an LAR-like algorithm for solving the problem for all values of the bound s.

We can generalize to the fused lasso support vector classifier by imposing constraints

$$\begin{aligned}\sum_{j=1}^{p}|\beta_j|&\leqslant s_1,\\ \sum_{j=2}^{p}|\beta_j-\beta_{j-1}|&\leqslant s_2.\end{aligned}\tag{16}$$

The complete set of constraints can be written as

$$\begin{pmatrix}1\\-a_0\\0\\0\\0\end{pmatrix}\leqslant\begin{pmatrix}I & y & y^{\mathrm{T}}X & 0 & 0 & 0 & 0\\ 0 & 0 & L & 0 & 0 & -I & I\\ 0 & 0 & I & -I & I & 0 & 0\\ 0 & 0 & 0 & e^{\mathrm{T}} & e^{\mathrm{T}} & 0 & 0\\ 0 & 0 & 0 & 0 & 0 & e^{\mathrm{T}} & e^{\mathrm{T}}\end{pmatrix}\begin{pmatrix}\xi\\ \beta_0\\ \beta\\ \beta^{+}\\ \beta^{-}\\ \theta^{+}\\ \theta^{-}\end{pmatrix}\leqslant\begin{pmatrix}\infty\\a_0\\0\\s_1\\s_2\end{pmatrix},\tag{17}$$

in addition to the bounds $\xi_i,\beta_j^{+},\beta_j^{-},\theta_j^{+},\theta_j^{-}\geqslant 0$. Since the objective function (15) is linear, this optimization is a linear (rather than quadratic) programming problem. Our implementation uses the SQOPT package again, as it handles both linear and quadratic programming problems.

We applied the fused lasso support vector classifier to the microarray leukaemia data. Using $s_1=2$ and $s_2=4$ gave a solution with 90 non-zero coefficients and 38 degrees of freedom. It produced one misclassification error in both tenfold cross-validation and the test set, making it competitive with the best classifiers from Table 4. Table 6 compares the signs of the fused lasso coefficients (rows) and the fused lasso support vector coefficients (columns). The agreement is substantial, but far from perfect.

One advantage of the support vector formulation is its fairly easy extension to multiclass problems: see for example Lee *et al.* (2002).

11. Discussion

The fused lasso seems a promising method for regression and classification, in settings where the features have a natural order.

One difficulty in using the fused lasso is computational speed. The timing results in Table 1 show that, when $p > 2000$ and $N > 200$, speed could become a practical limitation. This is especially true if five or tenfold cross-validation is carried out. Hot starts can help: starting with large values of (s_1, s_2), we obtain solutions for smaller values in a constant (short) time. (Initially we used *increasing* values of (s_1, s_2) because each solution is sure to be a feasible starting-point for the next values. However, with decreasing values of (s_1, s_2), SQOPT achieves feasibility quickly and has tended to be more efficient that way.)

The LAR algorithm of Efron *et al.* (2002) solves efficiently the entire sequence of lasso problems, for all values of the L_1-bound s_1. It does so by exploiting the fact that the solution profiles are piecewise linear functions of the L_1-bound, and the set of active coefficients changes in a predictable way. One can show that the fused lasso solutions are piecewise linear functions as we move in a straight line in the (λ_1, λ_2) plane (see Rosset and Zhu (2003)). Here (λ_1, λ_2) are the Lagrange multipliers corresponding to the bounds s_1 and s_2. Hence it might be possible to develop an LAR-style algorithm for quickly solving the fused lasso problem along these straight lines. However, such an algorithm would be considerably more complex than LAR, because of the many possible ways that the active sets of constraints can change. In LAR we can only add or drop a variable at a given step. In the fused lasso, we can add or drop a variable, or fuse or defuse a set of variables. We have not yet succeeded in developing an efficient algorithm for this procedure, but it will be a topic of future research.

Generalizations of the fused lasso to higher dimensional orderings may also be possible. Suppose that the features $x_{j,j'}$ are arranged on a two-way grid—e.g. in an image. Then we might constrain coefficients that are 1 unit apart in any direction, i.e. constraints of the form

$$\begin{gathered} \sum |\beta_{j,j'}| \leqslant s_1, \\ \sum_{|k-l|=1} |\beta_{j,k} - \beta_{j,l}| + \sum_{|k-l|=1} |\beta_{k,j} - \beta_{l,j}| \leqslant s_2. \end{gathered} \tag{18}$$

This would present interesting computational challenges, as the number of constraints is of the order p^2.

Acknowledgements

Tibshirani was partially supported by National Science Foundation grant DMS-9971405 and National Institutes of Health contract N01-HV-28183. Saunders was partially supported by National Science Foundation grant CCR-0306662 and Office of Naval Research grant N00014-02-1-0076. Philip Gill's continuing work on the quadratic programming solver SQOPT is also gratefully acknowledged.

References

Adam, B.-L., Qu, Y., Davis, J. W., Ward, M. D., Clements, M. A., Cazares, L. H., Semmes, O. J., Schellhammer, P. F., Yasui, Y., Feng, Z. and Wright, Jr, G. L. W. (2003) Serum protein fingerprinting coupled with a pattern-matching algorithm distinguishes prostate cancer from benign prostate hyperplasia and healthy mean. *Cancer Res.*, **63**, 3609–3614.

Boser, B., Guyon, I. and Vapnik, V. (1992) A training algorithm for optimal margin classifiers. In *Proc. Computational Learning Theory II, Philadelphia*. New York: Springer.

Chen, S. S., Donoho, D. L. and Saunders, M. A. (2001) Atomic decomposition by basis pursuit. *SIAM Rev.*, **43**, 129–159.

Donoho, D. and Johnstone, I. (1994) Ideal spatial adaptation by wavelet shrinkage. *Biometrika*, **81**, 425–455.

Efron, B., Hastie, T., Johnstone, I. and Tibshirani, R. (2002) Least angle regression. *Technical Report*. Stanford University, Stanford.

Geyer, C. (1996) On the asymptotics of convex stochastic optimization. *Technical Report*. University of Minnesota, Minneapolis.

Gill, P. E., Murray, W. and Saunders, M. A. (1997) Users guide for SQOPT 5.3: a Fortran package for large-scale linear and quadratic programming. *Technical Report NA 97-4*. University of California, San Diego.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. and Lander, E. (1999) Molecular classification of cancer: class discovery and class prediction by gene expression monitoring. *Science*, **286**, 531–536.

Hastie, T., Tibshirani, R. and Friedman, J. (2001) *The Elements of Statistical Learning; Data Mining, Inference and Prediction*. New York: Springer.

Hoerl, A. E. and Kennard, R. (1970) Ridge regression: biased estimation for nonorthogonal problems. *Technometrics*, **12**, 55–67.

Knight, K. and Fu, W. (2000) Asymptotics for lasso-type estimators. *Ann. Statist.*, **28**, 1356–1378.

Land, S. and Friedman, J. (1996) Variable fusion: a new method of adaptive signal regression. *Technical Report*. Department of Statistics, Stanford University, Stanford.

Lee, Y., Lin, Y. and Wahba, G. (2002) Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data. *Technical Report*. University of Wisconsin, Madison.

Petricoin, E. F., Ardekani, A. M., Hitt, B. A., Levine, P. J., Fusaro, V., Steinberg, S. M., Mills, G. B., Simone, C., Fishman, D. A., Kohn, E. and Liotta, L. A. (2002) Use of proteomic patterns in serum to identify ovarian cancer. *Lancet*, **359**, 572–577.

Rosset, S. and Zhu, J. (2003) Adaptable, efficient and robust methods for regression and classification via piecewise linear regularized coefficient paths. Stanford University, Stanford.

Rosset, S., Zhu, J. and Hastie, T. (2004) Boosting as a regularized path to a maximum margin classifier. *J. Mach. Learn. Res.*, **5**, 941–973.

Stein, C. (1981) Estimation of the mean of a multivariate normal distribution. *Ann. Statist.*, **9**, 1131–1151.

Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B, **58**, 267–288.

Tibshirani, R., Hastie, T., Narasimhan, B. and Chu, G. (2001) Diagnosis of multiple cancer types by shrunken centroids of gene expression. *Proc. Natn. Acad. Sci. USA*, **99**, 6567–6572.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J. and Knight, K. (2004) Sparsity and smoothness via the fused lasso. *Technical Report*. Stanford University, Stanford.

Vapnik, V. (1996) *The Nature of Statistical Learning Theory*. New York: Springer.

Wold, H. (1975) Soft modelling by latent variables: the nonlinear iterative partial least squares (NIPALS) approach. In *Perspectives in Probability and Statistics, in Honor of M. S. Bartlett*, pp. 117–144.

Zhu, J., Rosset, S., Hastie, T. and Tibshirani, R. (2003) L1 norm support vector machines. *Technical Report*. Stanford University, Stanford.