Sparsity and smoothness via the fused lasso

Robert Tibshirani, Michael Saunders,
Saharon Rosset, and Ji Zhu
Stanford University

August 5, 2003

Abstract

The lasso (Tibshirani 1996) 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, that is, having many coefficients equal to zero. In this paper we propose the "fused lasso", a generalization of the lasso designed for problems with features that come in a natural order. The fused lasso penalizes both the L_1 norm of the coefficients and their successive differences. Thus it encourages both sparsity of the coefficients and sparsity of their differences, that is, local constancy of the coefficient profile. The fused lasso is especially useful when the number of features $p \gg N$, 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.

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 zero or one, representing two classes like "healthy" and "diseased". We also assume that the x_{ij} are realizations of features X_j that are in a natural order $X_1, X_2, \ldots X_p$. 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*

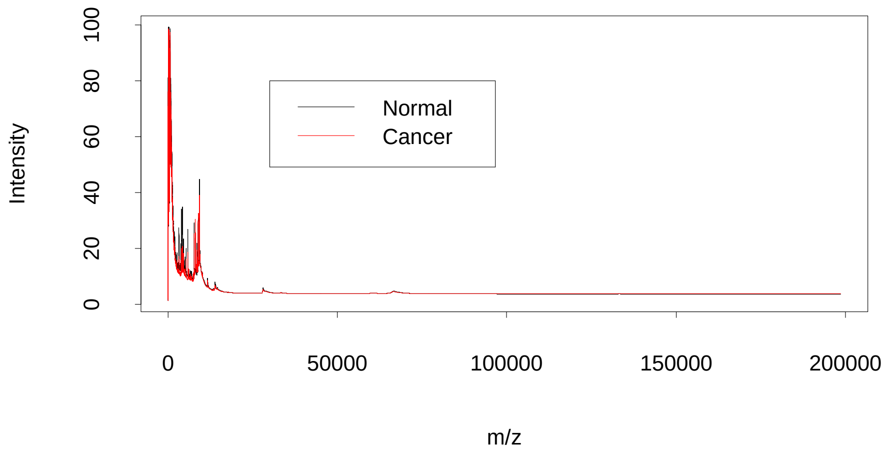

Figure 1: *Protein mass spectroscopy data: average profiles from normal and prostate cancer patients.*

flight values t_j. Time of flight is related to the mass over charge ratio (m/z) of the constituent proteins in the blood. Figure 1 shows an example taken from Adam et al. (2003). It shows the average spectra for healthy patients and those with prostate cancer. There are 48,538 m/z sites in total.. The full dataset consists of 157 healthy patients and 167 with cancer, and the goal is to find m/z sites that discriminate between the two groups. There has been much interest in this problem in the past few years, see e.g. Petricoin et al. (2002) and Adam et al. (2003).

In other examples, the order of the features may not be fixed apriori 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. In section 4 we relate the fused lasso to soft-threshold methods and wavelets. Degrees of freedom of the fused lasso fit are discussed in section 5. A protein mass spectroscopy dataset on prostate cancer in analyzed in section 6, while section 7 carries out a simulation study. application of the method to unordered features is discussed in Section 8 and illustrated on a microarray dataset in section 8.1. The hinge loss function and support vector classifiers are addressed in Section 9.

2 The lasso and fusion

We begin with a standard linear model

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

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

The lasso (Tibshirani 1996) finds the coefficients β_j satisfying

$$\hat{\beta} = \text{argmin} \sum_i (y_i - \sum_j x_{ij}\beta_j)^2 \quad \text{subject to} \sum_j |\beta_j| \leq s \tag{2}$$

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, that is, some components that are exactly zero. 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 (2) are convex, the problem can be solved even for large p (e.g. $p = 40,000$) by quadratic programming methods. In contrast, best subsets selection is not a convex problem and hence is prohibitive computationally for p as small as 50. We discuss computation in detail in section 3.

The lasso can be applied in the present setting, even if $p > N$, and it has a unique solution. An interesting property of the solution is the fact that the number of non-zero coefficients is at most $\min(N, p)$. Thus if $p = 40,000$ and $N = 100$, at most 100 coefficients in the solution will be non-zero. The "basis pursuit" signal estimation method of Chen et al. (1998) uses the same idea as the lasso, but applied in the wavelet domain.

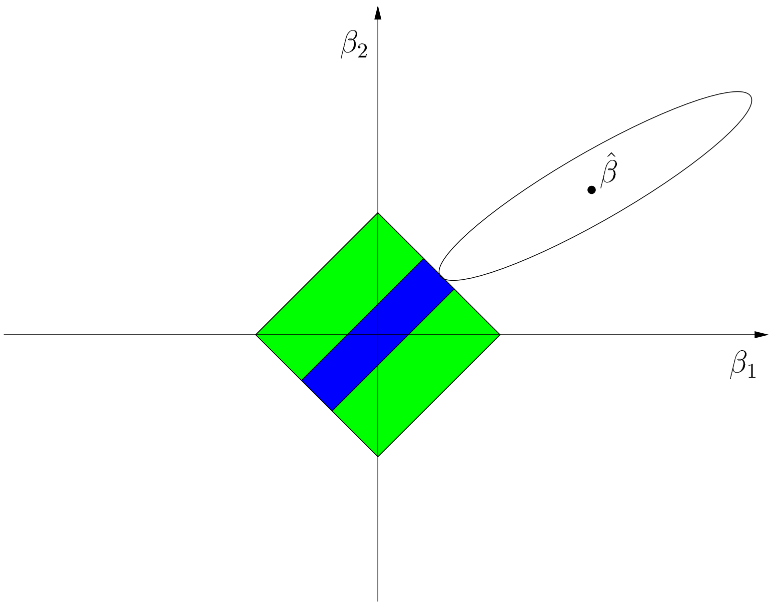

Figure 2: *Schematic of fused lasso. We seek the first time that the contours of the sum of squares loss function (ellipse) that satisfies* $\sum_j |\beta_j| = s_1$ *(green diamond) and* $\sum_j |\beta_j - \beta_{j-1}| = s_2$ *(blue slab).*

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

$$\hat{\beta} = \text{argmin} \sum_i (y_i - \sum_j x_{ij}\beta_j)^2$$
$$\text{subject to} \sum_{j=1}^{p} |\beta_j| \le s_1 \quad \text{and} \quad \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \le s_2 \tag{3}$$

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

Figure 3 illustrates these ideas on a simulated example. There are $p = 100$ predictors and $N = 20$ samples. The data were generated from a model $y_i = \sum_j x_{ij}\beta_j + \epsilon_i$ where the x_{ij} are standard Gaussian, $\epsilon_i \sim N(0, \sigma^2)$ with $\sigma = .75$, and there are 3 blocks of consecutive non-zero β_j's shown by the black points in each of the panels. The top left panel shows the univariate regression coefficients (red) and a soft-thresholded version of them (green). The top right panel shows the lasso solutions (red), using $s_1 = 35.6, s_2 = \infty$. The bottom left panel show the fusion estimate, using $s_1 = \infty$ and $s_2 = 26$. These values of s_1 and s_2 were the ones that minimized the estimated test set error. Finally the bottom right panel shows the fused lasso, using $s_1 = \sum_j |\beta_j|$ and $s_2 = \sum_j |\beta_j - \beta_{j-1}|$, β being the true set of ceofficients. The fused lasso does the best job in estimating the true underlying coefficients. However the fusion method (bottom left panel) performs as well as the fused lasso does in this example.

Figure 4 shows another example, with the same setup as in Figure 3 except that $\sigma = .05$ 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, s_2 were chosen in each case to minimize prediction error. The lasso performs poorly; fusion captures the plateau but does not clearly isolate the peak at $m/z = 10$. The fused does a good job overall.

3 Computational approach

3.1 Fixed s_1, s_2

Criterion (3) leads to a quadratic programming problem. For large p, the problem may be 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. (1999), which is designed for quadratic programming problems with sparse linear constraints.

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

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

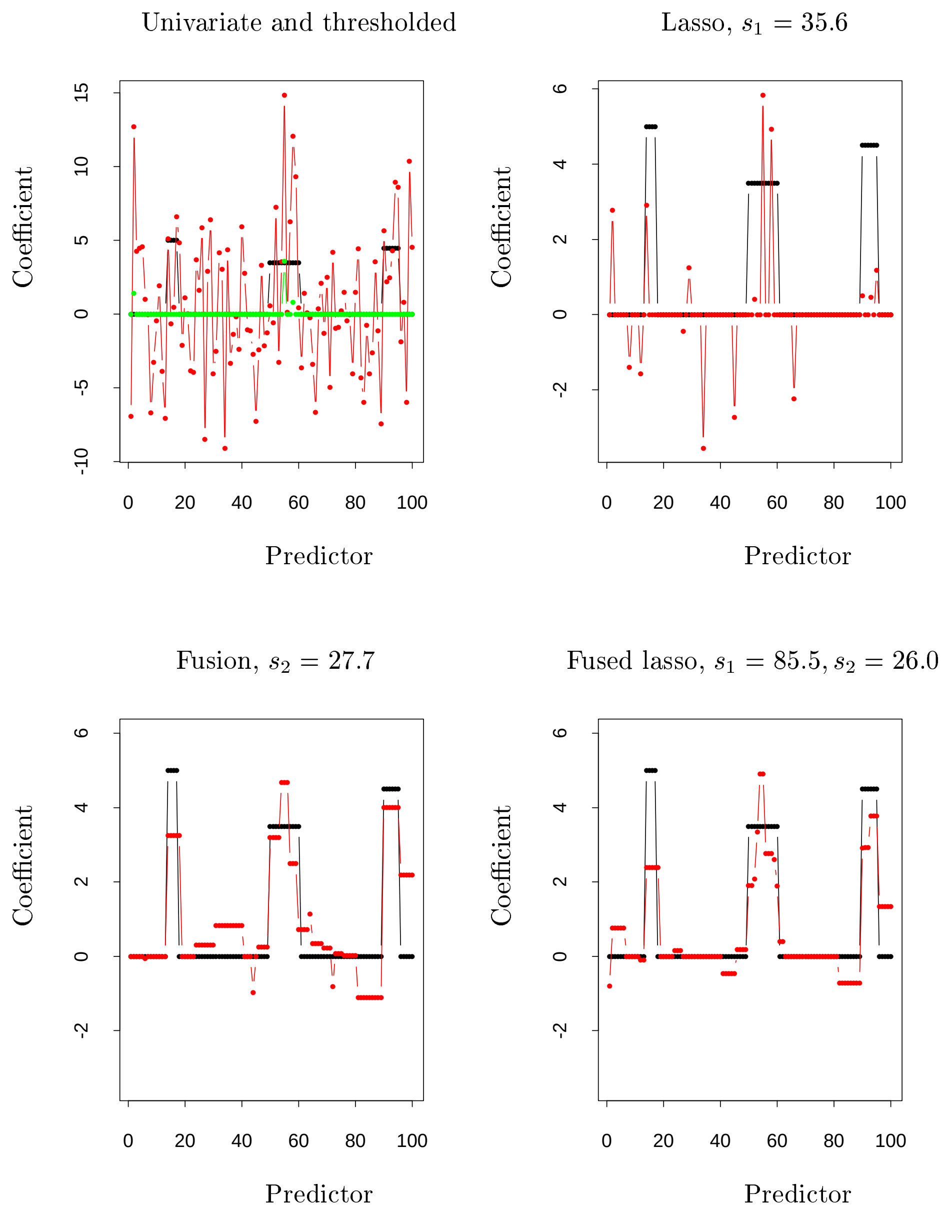

Figure 3: *Simulated example, with $p = 100$ predictors having coefficients shown by the black lines in each panel. The top left panel shows the univariate regression coefficients (red) and a soft-thresholded version of them (green). The top right panel shows the lasso solutions (red), using $s_1 = 35.6, s_2 = \infty$. The bottom left panel show the fusion estimate, using $s_1 = \infty$ and $s_2 = 26$. These values of s_1 and s_2 were the ones that minimized the estimated test set error. Finally the bottom right panel shows the fused lasso, using $s_1 = \sum_j |\beta_j|$ and $s_2 = \sum_j |\beta_j - \beta_{j-1}|$, β being the true set of ceofficients.*

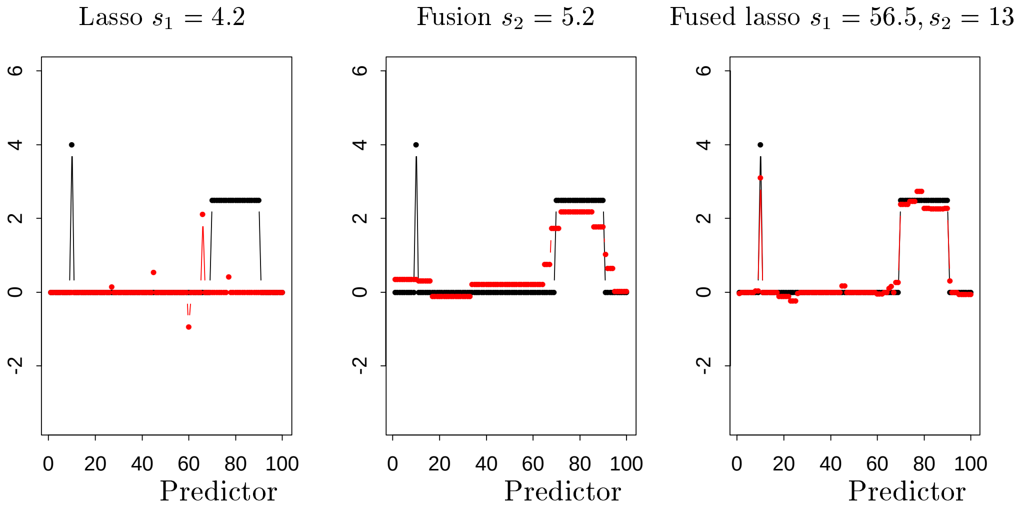

Figure 4: *Simulated example with only two areas of non-zero coefficients (black points and lines). Red points are the estimated coefficients from each method.*

$$\hat{\beta} = \quad \operatorname{argmin}(y - X\beta)^T(y - X\beta) \tag{4}$$

$$\text{subject to} \quad \begin{pmatrix} -a_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} L & 0 & 0 & -I & I \\ I & -I & I & 0 & 0 \\ 0 & e^T & e^T & 0 & 0 \\ 0 & 0 & 0 & e^T & e^T \end{pmatrix} \begin{pmatrix} \beta \\ \beta^+ \\ \beta^- \\ \theta^+ \\ \theta^- \end{pmatrix} \leq \begin{pmatrix} a_0 \\ 0 \\ s_1 \\ s_2 \end{pmatrix}, \tag{5}$$

in addition to the non-negativity constraints $\beta_j^+, \beta_j^-, \theta_j^+, \theta_j^- \geq 0$. The big matrix is of dimension $2p+2$ by $5p$, 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 $|\beta_1|$.

The `sqopt` package requires the user to write a procedure that computes $X^T X v$ for coefficient p-vectors v that are under consideration. For many choices of the bounds s_1, s_2, the vector v is very sparse and and hence $X^T X v$ can be efficiently computed. The algorithm is also well suited for "warm starts": starting at a solution for a given s_1, s_2, the solution for a nearby values of these bounds can be found relatively quickly.

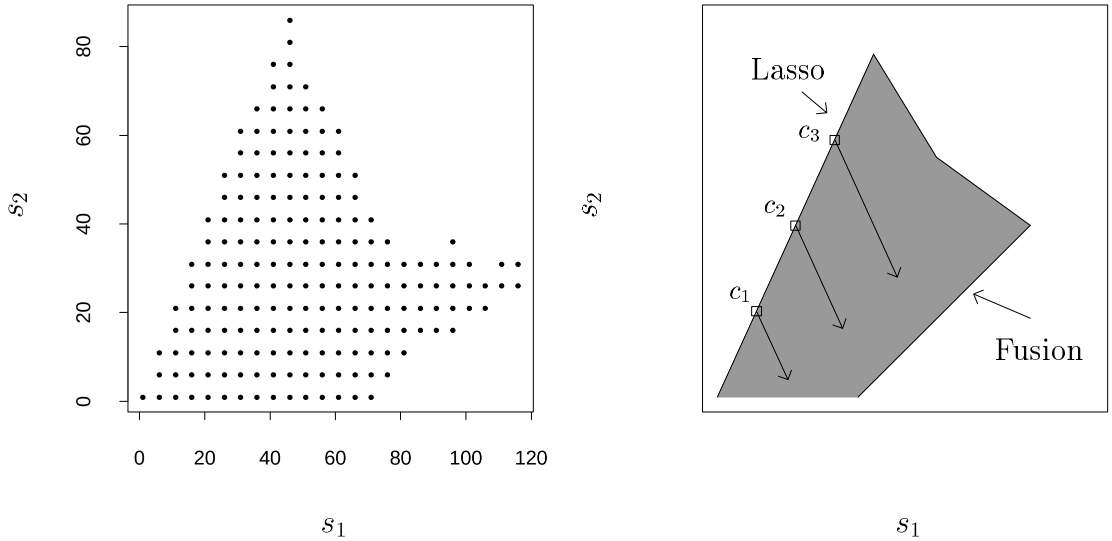

Figure 5: *Simulated example of Figure 3. Left panel shows attainable values of bounds s_1 and s_2. Right panel shows a schematic of the search process for the fused lasso, described in the text.*

3.2 Search strategy

For moderate-sized problems ($p \simeq 1000, N \simeq 100$ say), the above procedure is sufficiently fast that it can be applied over a grid of s_1, 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 using the LAR (least angle regression) 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, that is the solution vector satisfies both $\sum_j |\hat{\beta}_j| = s_1$ and $\sum_j |\hat{\beta}_j - \beta_{j-1}| = s_2$. The left panel of Figure 5 shows the attainable values for our simulated data example.

The right panel of Figure 5 is a schematic of the search strategy. Using the LAR procedure as above, we obtain solutions for bounds $(s_1(i), \infty)$, where $s_2(i)$ is the bound giving a solution with i degrees of freedom. [We discuss the "degrees of freedom" of the fused lasso fit in Section 5]. We use the lasso sequence of solutions and cross-validation or a test set, to estimate an optimal

p	N	Start	Time
100	20	Cold	.09s
500	20	Cold	1.0s
1000	20	Cold	2.0s
1000	200	Cold	30.4s
2000	200	Cold	2.0min
2000	200	Warm	16.6s

Table 1: *Timings for typical runs of fused lasso program*

degrees of freedom $\hat{\imath}$. Now let $s_{2max}(s_1(\hat{\imath})) = \sum_j |\hat{\beta}_j(s_1(\hat{\imath})) - \hat{\beta}_{j-1}(s_1(\hat{\imath}))|$. This is the largest value of the bound s_2 at which it affects the solution. The point c_2 in the Figure is $[s_1(\hat{\imath}), s_{2max}(s_1(\hat{\imath}))]$. 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 fuse the solutions from those points. The particular direction $(1, -2)$ was chosen by empirical experimentation. We are typically not interested in solutions near the pure fusion model (lower right boundary), and this search strategy tries to cover 9roughly) the potentially useful values of (s_1, s_2). This strategy is used in the real examples and simulation study, discussed later in the paper.

Table 3.2 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 10.

4 Soft-thresholding and wavelets

4.1 Soft-thresholding estimators

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

$$\hat{\beta}_j(\gamma_1) = \text{sign}(\tilde{\beta}_j) \cdot (|\tilde{\beta}_j| - \gamma_1)_+ \tag{6}$$

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

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 $\theta = L\beta$, $Z = XL^{-1}$. Note that L^{-1} is a lower-triangular matrix of ones, 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

$$\hat{\theta}_j(\gamma_2) = \text{sign}(\tilde{\theta}_j) \cdot (|\tilde{\theta}_j| - \gamma_2)_+, \tag{7}$$

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

Of course we can't have both $X^T X = I$ and $X^T X = L^T L$ 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 $U^T U = I$, and assume that the full least squares estimates $\beta' = (X^T X)^{-1} X^T y$ are non-decreasing: $0 \leq \beta'_1 \leq \beta'_2 ... \leq \beta'_p$. Finally, we set $s_1 = s_2 = s$. Then the fused lasso solution soft-thresholds the full least squares estimates β' from the right:

$$\hat{\beta} = (\beta'_1, \beta'_2, \ldots \beta'_k, \lambda, 0, 0, \ldots 0) \tag{8}$$

where $\sum_1^k \beta'_j + \lambda = s$. However this setup does not seem to be very useful in practice, as its assumptions are quite unrealistic.

4.2 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. 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 or by soft-thresholding or by lasso, giving $\tilde{\gamma}$. Finally we map back to get $\tilde{\beta} = W\tilde{\gamma}$. Note that soft-thresholding implicitly assumes that the Z basis is orthonormal, that is $Z^T Z = I$. But this is equivalent to $X^T X = WW^T$. For orthonormal Haar wavelets, $WW^T = I$, so $Z^T Z = I$ implies $X^T X = I$.

This procedure seeks a sparse representation of the βs in the transformed space. In contrast, the lasso and simple soft-thresholded estimates (6) 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

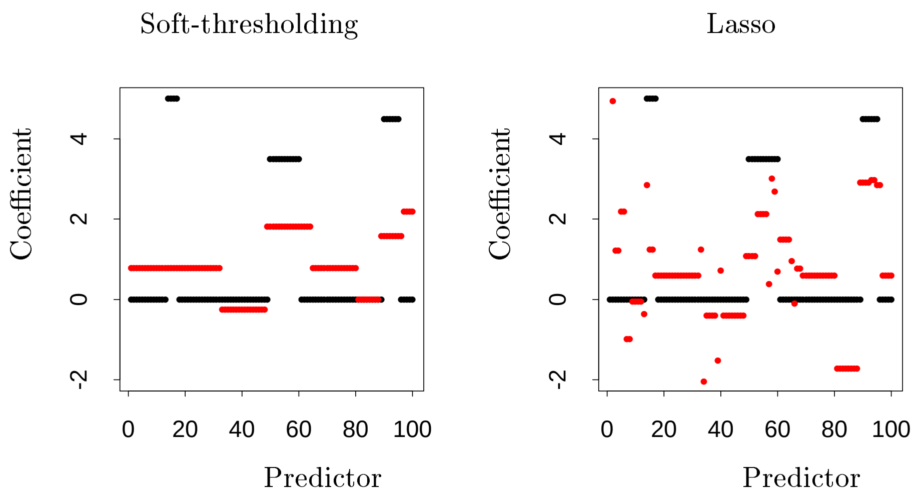

Figure 6: *Simulated example of Figure 3. Left panel: true coefficients (black), and estimated coefficients (red) obtained from transforming to a Haar wavelet basis, thresholding, and transforming back. Right panel: same procedure was carried out, except that the Lasso was applied to the Haar coefficients (rather than soft-thresholding).*

does not assume orthogonality, since this cannot hold simultaneously in both representations. The price for this ambition is an increased computational burden.

Figure 6 shows the results of applying soft-thresholding (left panel) or the lasso (right panel) in the space of Haar wavelets coefficients, and then transforming back to the original space. For soft-thresholding, we used the level-dependent threshold $\sigma\sqrt{2\log N_j}$ where N_j is the number of wavelet coefficients at the given scale, and σ was chosen to minimize test error (see e.g. Donoho & Johnstone (1994)). For the lasso, we chose the bound s_1 to minimize 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 nonzero coefficients are not situated near a power of two along the feature axis, the wavelet basis will have difficulty representing it.

5 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, s_2 are varied. Efron et al. (2002) consider a definition of degrees of freedom using the formula of Stein (1981):

$$\mathrm{df}(\hat{y}) = \frac{1}{\sigma^2}\sum_{i=1}^{N}\mathrm{cov}(y_i, \hat{y}_i). \tag{9}$$

Here 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 ($X^TX = I$), the lasso estimators are simply the soft-threshold estimates (6), and Efron et al. (2002) show that the degrees of freedom equals the number of non-zero coefficients. They also prove 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

$$\frac{1}{\sigma^2}\sum_{i=1}^{N}\mathrm{cov}(y_i, g_i) = \mathrm{E}\left(\sum_i \partial g(y)/\partial y_i\right) \tag{10}$$

Here $y = (y_1, y_2, \ldots y_N)$ 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 (6), $g(y)$ equals $\mathrm{sign}(y_i)(|y_i] - \gamma_1)$. The derivative $\partial g(y)/\partial y_i$ equals 1 if the ith component is non-zero, and zero otherwise. Hence the degrees of freedom is the number of non-zero coefficients.

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

$$\mathrm{df}(\hat{y}) = \#\{\text{nonzero coefficient blocks in } \hat{\beta}\}. \tag{11}$$

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

$$\mathrm{df}(\hat{y}) = p - \#\{\beta_j = 0\} - \#\{\beta_j - \beta_{j-1} = 0, \beta_j, \beta_{j-1} \neq 0\}. \tag{12}$$

It is easy to see that these two definitions the same. Furthermore, the objective function can be made zero when $\mathrm{df}(\hat{y}) \geq \min(N, p)$, and hence $\min(N, p)$

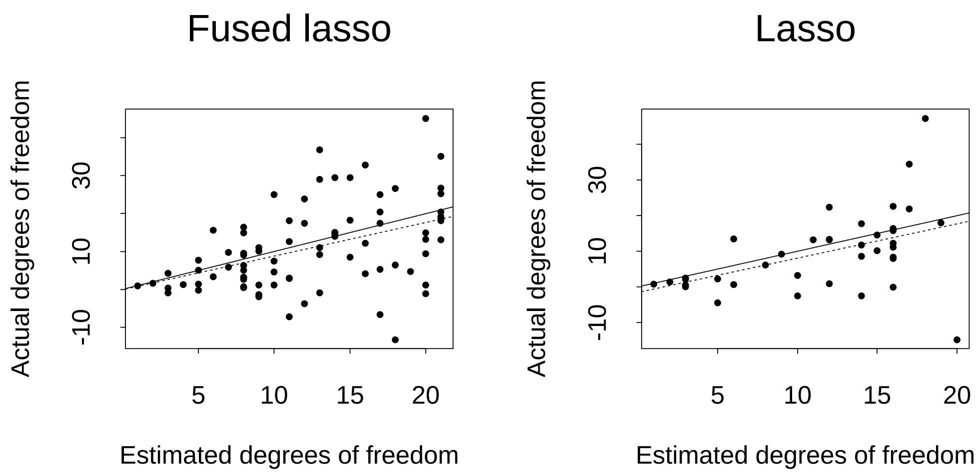

Figure 7: *Simulated example: actual and estimated degrees of freedom for fused lasso and lasso. In each panel the solid line isthe* 45^o *line and the broken line is the least squares regression fit.*

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 (10) in scenarios (6), (7), and (8).

Figure 7 compares the estimated and actual degrees of freedom for the fused lasso and lasso. The approximation for the fused lasso is fairly crude, but not much worse than that for the lasso.

6 Analysis of prostate cancer data

As mentioned in the introduction, this dataset consists of 48,538 measurements on 324 patients— 157 healthy patients and 167 with cancer. The average profiles (centroids) are shown in in Figure 1. We randomly created training and test sets of size 216 and 108 patients respectively. To make computations manageable, we sampled every 20th site, giving a total of 2427 sites. [Alternatively, one could average the data in blocks of 20]. The results of various methods are shown in Table 2. Nearest shrunken centroids (Tibshirani et al. 2001) is essentially equivalent, in this two-class setting, to soft-thresholding of the univariate regression coefficients.

Method	Test errors/108	df	# sites	s_1	s_2
Nearest shrunken centroids	26		169		
Lasso	6	116	116	144	262
Fusion	19	168	171	175	200
Fused Lasso	6	122	344	184	222

Table 2: *Prostate data results*

The lasso and fused lasso both perform well, with error rate about the same as that reported by the authors using a decision tree algorithm. Figure 8 shows the non-zero coefficients from the two methods, with the (standardized) difference between the centroids superimposed to aid in interpretation. We see that the lasso focuses almost entirely on low m/z sites, while the fused lasso spreads out more evenly, and suggests some higher m/z that may characterize the cancer profiles. In contrast, the authors' analysis gave no sites past m/z values of 10,000. 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 leukemia data in section 8.1.

Analysis of peaks

Figure 9 shows the centroids of each group, marking the sites having lasso and fused lasso coefficients > 1 in absolute value. We see that many of the selected sites do not occur near a peak of either the control or cancer centroid. All else equal, it would be preferable for the chosen sites to occur near peaks, as that would suggest an effect due to an over or under-abundance of a protein at that m/z value.

We applied a simple peak-finding algorithm to each of the two centroids, defining a peak to be a lcoal minimum with lower values at each of 5 neighbors on the left and right. This produced 170 peaks. Then we applied the lasso and fused lasso to these 170 sites, with all other coefficients set to zero. For the fused lasso, this is done simply by defining upper bounds of zero for the corresponding positive and negative components β_j^+ and β_j^-. The results are given in Table 3. The non-zero fused lasso coefficients are shown in Figure 9. Restricting the model to local peaks seems to increase the error rate. This suggests that informative proteins may be super-imposed, so that some discriminating information occurs on the "shoulders" of the profiles.

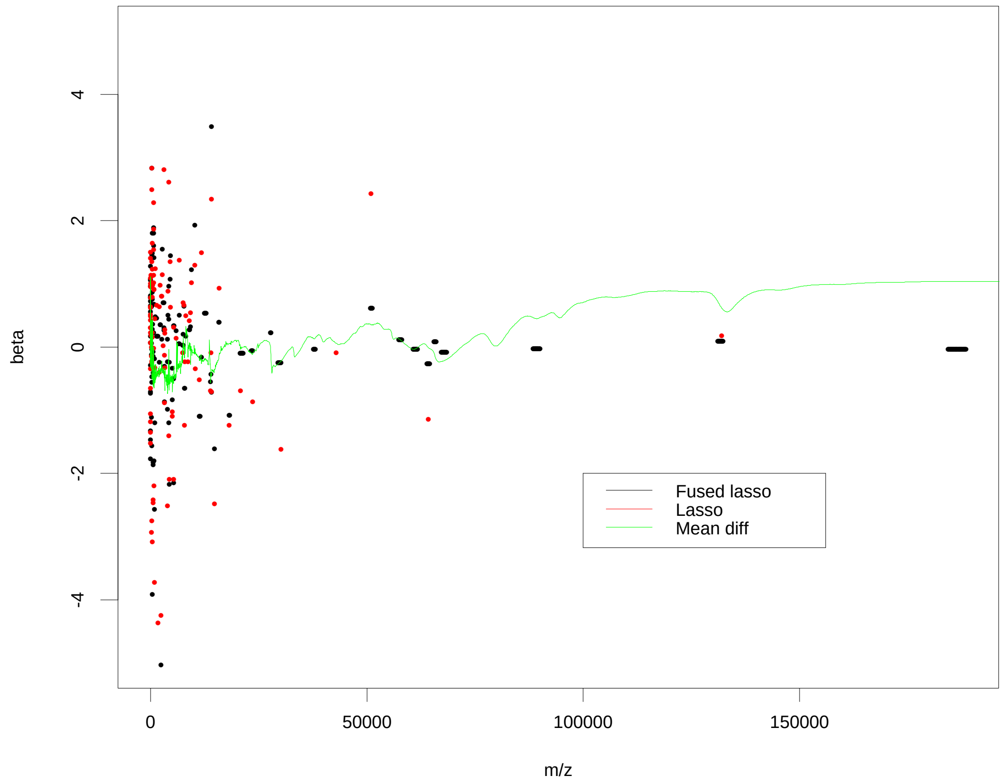

Figure 8: *Results for prostate cancer example. Lasso and fused lasso non-zero coefficients are shown in red and black respectively. Green curve shows the standardized difference between the means, for reference.*

Method	Test errors/108	df	# sites	s_1	s_2
Lasso	15	58	58	75	137
Fused Lasso	13	91	94	338	636

Table 3: *Prostate data results: analysis of peaks*

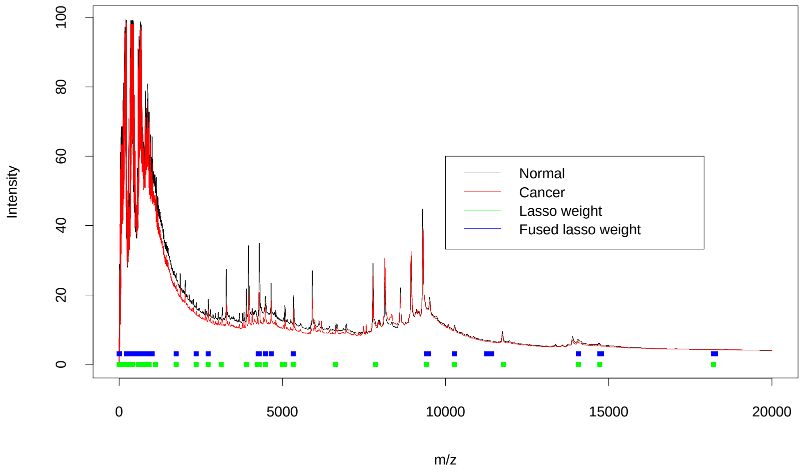

Figure 9: *Results for prostate cancer example. Positions of largest non-zero lasso and fused lasso coefficients are shown in green and blue respectively.*

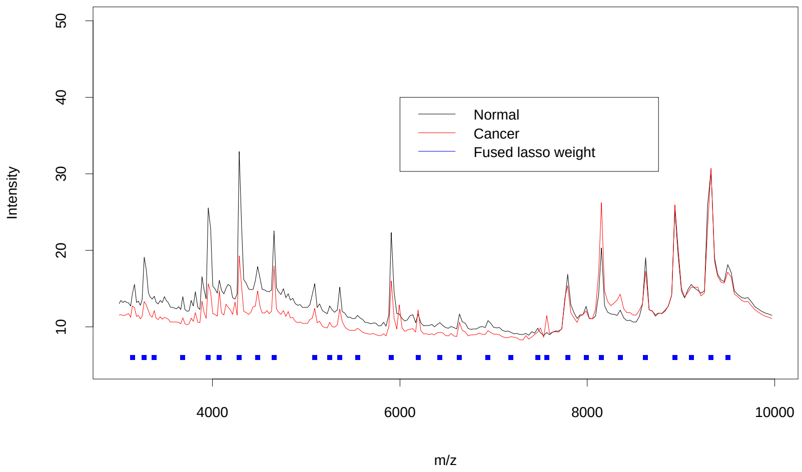

Figure 10: *Results for prostate cancer example, analysis of peaks. Positions of non-zero fused lasso coefficients are shown in blue; for clarity only a portion of the horizontal range is shown.*

Method	Test Error (se)	Sensitivity (se)	Specificity (se)
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)

Table 4: *Results of simulation study*

7 A simulation study

We carried out a small simulation study to compare the performance of the lasso and 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 described in the previous section. We also used a random subset of 100 of the patients, to keep the feature/ sample size ratio near a realistic level. We then generated coefficient vectors β by choosing 1 to 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

$$y = X\beta + Z; \quad 2.5 \cdot Z \sim N(0,1). \tag{13}$$

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

For each dataset, we found the lasso solution with minimum test error. We then used the search strategy outlined in section (3) for the fused lasso. Table 4 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, s_2.

We see that the fused lasso slightly improves upon 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. Note that even with the true s_1, s_2 bounds, the fused lasso detects less than half of the true non-zero coefficients. This demonstrates the inherent difficulty of problems having $p \gg N$.

8 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 pre-specified order: for example, genes in a microarray experiments. There are at least two ways to apply the fused lasso in this case. First, one can estimate an order for the features, using for example multidimensional scaling or hierarchical clustering. The latter is commonly used for creating heatmap displays of microarray data.

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

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

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

8.1 Leukemia classification using microarrays

These data were introduced in Golub et al. (1999). There are 7129 genes and 38 samples: 27 in class 1 (Acute Lymphocytic Leukemia) and 11 in class 2 (Acute Mylogenous Leukemia). In addition there is a test sample of size 34. The prediction results are shown in Table 5.

The first two lines are based on all 7129 genes. Golub *et. al.*'s procedure is similar to nearest shrunken centroids, but 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 fit by optimizing the tuning parameters using cross-validation and then applying these values to the test set. The pure fusion estimate (6) did poorly in test error: this error never dropped below 3 for any value of the bound s_2.

Method	10-fold CV error	Test error	# genes
(1) Golub (50 genes)	3/38	4/34	50
(2) Nearest shrunken centroid (21 genes)	1/38	2/34	21
(3) Lasso 37 df ($s_1 = .65, s_2 = 1.32$)	1/38	1/34	37
(4) Fused lasso 38 df ($s_1 = 1.08, s_2 = .71$)	1/38	2/34	135
(5) Fused lasso 20 df ($s_1 = 1.35, s_2 = 1.01$)	1/38	4/34	737
(6) Fusion 1 df	1/38	12/34	975

Table 5: *Results for leukemia microarray example*

We see that in line (4) fusing the lasso solution gives about the same error rate, using about four times as many genes. Further fusion in line (5) seems to increase the test error rate. Table 6 shows the estimated coefficients for the Lasso and Fused Lasso solution (4). We see that in many cases, the fusion process has spread out the coefficient of a non-zero lasso coefficient onto adjacent genes.

9 Hinge loss

For two-class problems the maximum margin approach 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 e.g. Hastie et al. (2001), chapter 11). We minimize

$$J(\beta_0, \beta, \xi) = \sum_{j=1}^{n} \xi_j \tag{14}$$

subject to $y_i(\beta_0 + \beta^T x_i) \geq 1 - \xi_i$, $\xi_i \geq 0$ for all i. The original support vector classifier includes an L_2 constraint $\sum_{j=1}^{p} \beta_j^2 \leq s$. Recently there has been interest in the L_1 constrained (Lasso) support vector classifier. Zhu et al. (2003) develop a LARS-like algorithm for solving the problem for all values of the bound s.

We can generalize to the "fused lasso" support vector classifier by impos-

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
54	0.00000	-0.00830	570	0.00000	-0.00005	839	0.00000	-0.00408
55	-0.02027	-0.00830	571	0.00000	-0.00005	840	0.00000	-0.00408
			572	-0.00764	-0.00005			
59	-0.01383	0.00000				881	0.00000	0.00680
			579	0.00000	-0.00119	882	0.00000	0.00680
74	-0.00014	0.00000	580	0.00000	-0.00119	883	0.00000	0.00680
75	0.00000	-0.00537	581	0.00000	-0.00119	884	0.00000	0.00030
76	0.00000	-0.00537	582	0.00000	-0.00119	885	0.00629	0.00030
			583	-0.00220	-0.00119			
143	0.03479	0.01008				906	0.02490	0.02088
144	0.00000	0.01008	616	0.01102	0.01743	907	0.00000	0.02088
145	0.00000	0.01008	617	0.00495	0.01743			
146	0.00000	0.01008				926	0.00000	0.00146
147	0.00000	0.01008	670	-0.00496	0.00000	927	0.00000	0.00146
						928	0.00000	0.00146
171	0.00000	0.00664	675	0.00000	0.00000	929	0.00000	0.00146
172	0.00057	0.00664	676	0.00000	0.00000	930	0.00000	0.00146
173	0.00000	0.00664	677	0.00000	0.00000	931	0.00000	0.00146
			678	0.00000	0.00000	932	0.00000	0.00650
176	0.00256	0.00000	679	0.00000	0.00000	933	0.00000	0.00650
			680	0.00000	0.00000	934	0.00000	0.00650
192	-0.00313	0.00000	681	0.00000	0.00000	935	0.00000	0.00650
			682	0.00000	0.00000			
201	0.00000	0.00100	683	0.00000	0.00000	939	0.14107	0.05892
202	0.00179	0.00100	684	0.00000	0.00000	940	0.00000	0.04174
203	0.00000	0.00100						
204	0.00000	0.00100	699	0.01276	0.00000	949	0.00000	0.01018
						950	0.00000	0.01018
234	0.00000	0.00689	706	0.00000	0.00103	951	0.00000	0.01018
235	0.00000	0.00689	707	0.00000	0.00103	952	0.00000	0.01018
236	0.00000	0.00689				953	0.02783	0.03009
237	0.00000	0.00689	712	0.00205	0.00000			
						967	0.00000	-0.01177
248	0.00000	0.00164	716	0.00552	0.00000	968	-0.02327	-0.01177
249	0.00000	0.00164				969	0.00000	0.00681
250	0.00000	0.00164	718	0.02768	0.00000	970	0.00000	0.00681
251	0.00000	0.00164				971	0.00000	0.00681
			742	0.00000	0.00618	972	0.00000	0.00681
295	-0.03217	0.00000	743	0.00000	0.00618	973	0.00000	0.00681
			744	0.00000	0.00618	974	0.00000	0.00681
392	0.00000	-0.00510	745	0.00000	0.00878	975	0.00000	0.00681
393	0.00000	-0.00510	746	0.02103	0.00878			
394	0.00000	-0.00510				980	0.00000	0.00000
						993	0.00000	-0.00151
						994	0.00000	-0.00151

Table 6: *Leukemia data example: non-zero coefficients for lasso and fused lasso, with contiguous blocks delineated.*

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

Table 7: *Signs of fused lasso coefficients (rows) versus signs of fused lasso support vector coefficients (columns).*

ing constraints

$$\sum_{j=1}^{p} |\beta_j| \leq s_1; \quad \sum_{j=2}^{p} |\beta_j - \beta_{j-1}| \leq s_2 \tag{15}$$

The complete set of constraints can be written as

$$\begin{pmatrix} 1 \\ -a_0 \\ 0 \\ 0 \\ 0 \end{pmatrix} \leq \begin{pmatrix} I & y & y \cdot X & 0 & 0 & 0 & 0 \\ 0 & 0 & L & 0 & 0 & -I & I \\ 0 & 0 & I & -I & I & 0 & 0 \\ 0 & 0 & 0 & e^T & e^T & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & e^T & e^T \end{pmatrix} \begin{pmatrix} \xi \\ \beta_0 \\ \beta \\ \beta^+ \\ \beta^- \\ \theta^+ \\ \theta^- \end{pmatrix} \leq \begin{pmatrix} \infty \\ a_0 \\ 0 \\ s_1 \\ s_2 \end{pmatrix}, \tag{16}$$

in addition to the non-negativity constraints $\xi_i, \beta_j^+, \beta_j^-, \theta_j^+, \theta_j^- \geq 0$. Since the objective function (14) is linear, this optimization is a linear (rather than quadratic) programming problem. Our implementation uses the `sqopt` package as before; `sqopt` handles both LP and QP problems.

We applied the fused lasso support vector classifier to the microarray leukemia data. Using $s_1 = 2, s_2 = 4$ gave a solution with 90 non-zero coefficients, and 38 degrees of freedom. It produced one misclassification error in both tenfold cross-validation and test set, making it competitive with the best classifiers from Table 5. Table 7 compares the signs of the fused lasso coefficients (rows) and signs of 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 multi-class problems: see for example Lee et al. (2002).

10 Discussion

The fused lasso seems like 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 3.2 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.

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 & Zhu 2003)). Here (λ_1, λ_2) are the Lagrange multipliers corresponding to the bounds s_1, s_2. Hence it might be possible to develop a 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 one can only add or drop a variable at a given step. In the fused lasso, one can add or drop a variable, or fuse or de-fuse 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, for example in an image. Then we might constrain coefficients that are one unit apart in any direction, i.e. constraints of the form

$$\sum |\beta_{j,j'}| \leq s_1, \sum_{|k-\ell|=1} |\beta_{j,k} - \beta_{j,\ell}| + \sum_{|k-\ell|=1} |\beta_{k,j} - \beta_{\ell,j}| \leq s_2, \tag{17}$$

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

Acknowledgments

Tibshirani was partially supported by National Science Foundation Grant DMS-9971405 and National Institutes of Health Contract N01-HV-28183.

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. & 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 Research* **63(10)**, 3609–3614.

Boser, B., Guyon, I. & Vapnik, V. (1992), A training algorithm for optimal margin classifiers, *in* 'Proceedings of COLT II', Philadelphia, Pa.

Chen, S. S., Donoho, D. L. & Saunders, M. A. (1998), 'Atomic decomposition by basis pursuit', *SIAM Journal on Scientific Computing* pp. 33–61.

Donoho, D. & Johnstone, I. (1994), 'Ideal spatial adaptation by wavelet shrinkage', *Biometrika* **81**, 425–55.

Efron, B., Hastie, T., Johnstone, I. & Tibshirani, R. (2002), Least angle regression, Technical report, Stanford University.

Gill, P., Murray, W. & Saunders, M. (1999), Users guide for sqopt 5.3: a fortran package for large-scale linear and quadratic programming., Technical report, Stanford University.

Golub, T., Slonim, D., Tamayo, P., Huard, C., Gaasenbeek, M., Mesirov, J., Coller, H., Loh, M., Downing, J., Caligiuri, M., Bloomfield, C. & Lander, E. (1999), 'Molecular classification of cancer: class discovery and class prediction by gene expression monitoring', *Science* **286**, 531–536.

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

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

Lee, Y., Lin, Y. & Wahba, G. (2002), Multicategory support vector machines, theory, and application to the classification of microarray data and satellite radiance data, Technical report, Univ. of Wisconsin.

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

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

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

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

Tibshirani, R., Hastie, T., Narasimhan, B. & Chu, G. (2001), 'Diagnosis of multiple cancer types by shrunken centroids of gene expression', *Proc. Natl. Acad. Sci.* **99**, 6567–6572.

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

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