A SPARSE-GROUP LASSO

NOAH SIMON, JEROME FRIEDMAN, TREVOR HASTIE,
AND ROB TIBSHIRANI

Abstract. For high dimensional supervised learning problems, often using problem specific assumptions can lead to greater accuracy. For problems with grouped covariates, which are believed to have sparse effects both on a group and within group level, we introduce a regularized model for linear regression with ℓ_1 and ℓ_2 penalties. We discuss the sparsity and other regularization properties of the optimal fit for this model, and show that it has the desired effect of group-wise and within group sparsity. We propose an efficient algorithm to fit the model via accelerated generalized gradient descent, and extend this model and algorithm to convex loss functions. We also demonstrate the efficacy of our model and the efficiency of our algorithm on simulated data.

Keywords: penalize, regularize, regression, model, nesterov

1. Introduction

Consider the usual linear regression framework. Our data consists of an n response vector y, and an n by p matrix of features, X. In many recent applications we have $p >> n$: a case where standard linear regression fails. To combat this, Tibshirani [1996] regularized the problem by bounding the ℓ_1 norm of the solution. This approach, known as the lasso, minimizes

$$\frac{1}{2}\,||y - X\beta||_2^2 + \lambda||\beta||_1. \tag{1}$$

It finds a solution with few nonzero entries in β. Suppose, further, that our predictor variables were divided into m different groups— for example in gene expression data these groups may be gene pathways, or factor level indicators in categorical data. We are given these group memberships and rather than just sparsity in β we would like a solution which uses only a few of the groups. Yuan and Lin [2007] proposed the

group lasso criterion for this problem; the problem is

$$\min_\beta \frac{1}{2}\left|\left|y - \sum_{l=1}^{m} X^{(l)}\beta^{(l)}\right|\right|_2^2 + \lambda \sum_{l=1}^{m} \sqrt{p_l}||\beta^{(l)}||_2 \tag{2}$$

where $X^{(l)}$ is the submatrix of X with columns corresponding to the predictors in group l, $\beta^{(l)}$ the coefficient vector of that group and p_l is the length of $\beta^{(l)}$. This criterion exploits the non-differentiability of $||\beta^{(l)}||_2$ at $\beta^{(l)} = 0$; setting groups of coefficients to exactly 0. The sparsity of the solution is determined by the magnitude of the tuning parameter λ. If the size of each group is 1, this gives us exactly the regular lasso solution.
While the group lasso gives a sparse set of groups, if it includes a group in the model then all coefficients in the group will be nonzero. Sometimes we would like both sparsity of groups and within each group—for example if the predictors are genes we would like to identify particularly "important" genes in pathways of interest. Toward this end we focus on the "sparse-group lasso"

$$\min_\beta \frac{1}{2n}\left|\left|y - \sum_{l=1}^{m} X^{(l)}\beta^{(l)}\right|\right|_2^2 + (1-\alpha)\lambda \sum_{l=1}^{m} \sqrt{p_l}||\beta^{(l)}||_2 + \alpha\lambda||\beta||_1. \tag{3}$$

where $\alpha \in [0, 1]$ — a convex combination of the lasso and group lasso penalties ($\alpha = 0$ gives the group lasso fit, $\alpha = 1$ gives the lasso fit).

In this paper we discuss properties of this criterion, first proposed in our unpublished note, Friedman et al.. We extend it to logistic and Cox regression, and develop an algorithm to solve the original problem and extensions to other loss functions. Our algorithm is based on Nesterov's method for generalized gradient descent. By employing warm starts we efficiently solve the problem along a path of constraint values. We demonstrate the efficacy of our objective function and algorithm on real and simulated data, and we provide a publically available R implementation of our algorithm in the package `SGL`. This paper is the continuation of Friedman et al., a brief note on the criterion.

In Section 2 we develop the criterion and discuss some of its properties. We present the details of the algorithm we use to fit this model in Section 3. In Section 4 we extend this model to any log-concave likelihood in particular to logistic regression and the Cox proportional hazards model. In Section 5 we show the efficacy of our model and the efficiency of our algorithm on simulated data.

2. Criterion

Returning to the usual regression framework we have an n response vector y, and an n by p covariate matrix X broken down into m submatrices, $X^{(1)}, X^{(2)}, \ldots, X^{(m)}$, with each $X^{(l)}$ an n by p_l matrix, where p_l is the number of covariates in group l. We choose $\hat{\beta}$ to minimize

$$\frac{1}{2n}\left|\left|y - \sum_{l=1}^{m} X^{(l)}\beta^{(l)}\right|\right|_2^2 + (1-\alpha)\lambda\sum_{l=1}^{m}\sqrt{p_l}||\beta^{(l)}||_2 + \alpha\lambda||\beta||_1. \tag{4}$$

For the rest of the paper we will supress the $\sqrt{p_l}$ in the $\sum_{l=1}^{m}\sqrt{p_l}||\beta^{(l)}||_2$ penalty term for ease of notation. To add it back in, simply replace all future $(1-\alpha)\lambda$ by $\sqrt{p_k}(1-\alpha)\lambda$. One might note that this looks very similar to the elastic net penalty proposed by Zou and Hastie [2005]. It differs because the $||\cdot||_2$ penalty is not differentiable at $\mathbf{0}$, so some groups are completely zeroed out. However, as we show shortly, within each nonzero group it gives an elastic net fit (though with the $||\cdot||_2^2$ penalty parameter a function of the optimal $||\hat{\beta^{(k)}}||_2$).

The objective in (4) is convex, so the optimal solution is characterized by the subgradient equations. We consider these conditions to better understand the properties of $\hat{\beta}$. For group k, $\hat{\beta}^{(k)}$ must satisfy

$$\frac{1}{n}X^{(k)\top}r_{(-k)} = (1-\alpha)\lambda u + \alpha\lambda v$$

with $r_{(-k)}$ the partial residual of y, subtracting all group fits other than group k

$$r_{(-k)} = y - \sum_{l\neq k} X^{(l)}\hat{\beta}^{(l)}$$

and where u and v are subgradients of $||\hat{\beta}^{(k)}||_2$ and $||\hat{\beta}^{(k)}||_1$ respectively, evaluated at $\hat{\beta}^{(k)}$. So,

$$u = \begin{cases} \frac{\hat{\beta}^{(k)}}{||\hat{\beta}^{(k)}||_2} & \text{if } \hat{\beta}^{(k)} \neq \mathbf{0} \\ \in \{u : ||u||_2 \leq 1\} & \text{if } \hat{\beta}^{(k)} = \mathbf{0} \end{cases}$$

$$v_j = \begin{cases} \text{sign}\left(\hat{\beta}_j^{(k)}\right) & \text{if } \hat{\beta}_j^{(k)} \neq 0 \\ \in \{v_j : |v_j| \leq 1\} & \text{if } \hat{\beta}_j^{(k)} = 0 \end{cases}$$

With a little bit of algebra we see that the subgradient equations are satisfied with $\hat{\beta}^{(k)} = \mathbf{0}$ if

$$\left|\left|\mathrm{S}\left(X^{(k)\top}r_{(-k)}/n,\, \alpha\lambda\right)\right|\right|_2 \leq (1-\alpha)\lambda \tag{5}$$

with $S(\cdot)$ the coordinate-wise soft thresholding operator:

$$(S(z,\, \alpha\lambda))_j = \text{sign}(z_j)(|z_j| - \alpha\lambda)_+.$$

In comparison, the usual group lasso has $\hat{\beta}^{(k)} = \mathbf{0}$ if

$$\left|\left|X^{(k)\top} r_{(-k)}\right|\right|_2 \leq \lambda_2$$

Note that $r_{(-k)}$ differs between the group lasso and sparse-group lasso solutions. However, on a group sparsity level the two act similarly, though the sparse-group lasso adds univariate shrinkage before checking if a group is nonzero.

The subgradient equations can also give insight into the sparsity within a group which is at least partially nonzero. If $\beta^{(k)} \neq \mathbf{0}$ then the subgradient conditions for a particular $\beta_i^{(k)}$ become

$$\frac{1}{n} X_i^{(k)\top} \left(Y - \sum_{l=1}^{m} X^{(l)} \hat{\beta}^{(l)} \right) = (1-\alpha)\lambda \left(\frac{\hat{\beta}_i^{(k)}}{||\hat{\beta}^{(k)}||_2} \right) + \alpha\lambda v_i.$$

This is satisfied for $\hat{\beta}_i^{(k)} = 0$ if

$$|X_i^{(k)\top} r_{(-k,i)}| \leq n\alpha\lambda \tag{6}$$

with $r_{(-k,i)} = r_{(-k)} - \sum_{j \neq i} X_j^{(k)} \hat{\beta}^{(k)}$ the partial residual of y subtracting all other covariate fits, excluding only the fit of $X_i^{(k)}$. This is the same condition for a covariate to be inactive as in the regular lasso. Again however, the partial residuals of the optimal solution are different between the two models.

For $\beta_i^{(k)}$ nonzero, more algebra gives us that $\beta_i^{(k)}$ satisfies

$$\hat{\beta}_i^{(k)} = \frac{\text{S}\left({X_i^{(k)}}^\top r_{(-k,i)}/n,\, \alpha\lambda \right)}{{X_i^{(k)}}^\top X_i^{(k)}/n + (1-\alpha)\lambda/||\hat{\beta}^{(k)}||_2}. \tag{7}$$

These are elastic net type conditions as in Friedman et al. [2009]. Unlike the usual elastic net, the proportional shrinkage here is a function of the optimal solution, $\lambda_{\text{net},2} = (1-\alpha)\lambda/||\hat{\beta}^{(k)}||_2$. Formula (7) suggests a cyclical coordinate-wise algorithm to fit the model within group. We tried this algorithm in a number of incarnations and found it inferior in both timing and accuracy to the algorithm discussed in section 3. Puig et al. [2009] and Foygel and Drton [2010] fit the group lasso and sparse-group lasso respectively by explicitly solving for this parameter. This requires doing matrix calculations, which may be slow for larger group sizes, so we take a different approach.

From the subgradient conditions we see that this model promotes the desired sparsity pattern. Furthermore, it regularizes nicely within each group — giving an elastic net-like solution.

3. Algorithm

In this section we describe how to fit the sparse-group lasso using blockwise descent — to solve within each group we employ an accelerated generalized gradient algorithm with backtracking. Because our penalty is separable between groups, blockwise descent is guaranteed to converge to the global optimum.

3.1. **Within Group Solution.** We choose a group k to minimize over, and consider the other group coefficients as fixed — we can ignore penalties corresponding to coefficients in these groups. Our minimization problem becomes, find $\hat{\beta}^{(k)}$ to minimize

$$\frac{1}{2n}\left|\left|r_{(-k)} - X^{(k)}\beta^{(k)}\right|\right|_2^2 + (1-\alpha)\lambda||\beta^{(k)}||_2 + \alpha\lambda||\beta^{(k)}||_1 \tag{8}$$

We denote our unpenalized loss function by

$$\ell(r_{(-k)}, \beta) = \frac{1}{2n}\left|\left|r_{(-k)} - X^{(k)}\beta\right|\right|_2^2$$

Note, we are using β here to denote the coefficients in only group k. The modern approach to gradient descent is to consider it as a majorization minimization scheme. We majorize our loss function, centered at a point β_0 by

$$\ell(r_{(-k)}, \beta) \leq \ell(r_{(-k)}, \beta_0) + (\beta - \beta_0)^\top \nabla \ell(r_{(-k)}, \beta_0) + \frac{1}{2t}||\beta - \beta_0||_2^2 \tag{9}$$

where t is sufficiently small that the quadratic term dominates the Hessian of our loss; note, the gradient in $\nabla\ell(r_{(-k)}, \beta_0)$ is only taken over group k. Minimizing this function would give us our usual gradient step (with stepsize t) in the unpenalized case. Adding our penalty to (9) majorizes the objective (8).

$$M(\beta) = \ell(r_{(-k)}, \beta_0) + (\beta - \beta_0)^\top \nabla \ell(r_{(-i)}, \beta_0) + \frac{1}{2t}||\beta - \beta_0||_2^2 + (1-\alpha)\lambda||\beta||_2 + \alpha\lambda||\beta||_1.$$

Our goal now is to find $\tilde{\beta}$ to minimize $M(\cdot)$. Minimizing $M(\cdot)$ is equivalent to minimizing

$$\tilde{M}(\beta) = \frac{1}{2t}||\beta - (\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0))||_2^2 + (1-\alpha)\lambda||\beta||_2 + \alpha\lambda||\beta||_1. \tag{10}$$

Combining the subgradient conditions with basic algebra, we get that $\hat{\beta} = 0$ if

$$||S(\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0),\, t\alpha\lambda)||_2 \leq t(1-\alpha)\lambda$$

and otherwise $\hat{\beta}$ satisfies

$$\left(1 + \frac{t(1-\alpha)\lambda}{||\hat{\beta}||_2}\right)\hat{\beta} = S(\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0),\, t\alpha\lambda). \tag{11}$$

Taking the norm of both sides we see that

$$||\hat{\beta}||_2 = \left(||S(\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0),\, t\alpha\lambda)||_2 - t(1-\alpha)\lambda\right)_+.$$

If we plug this into (11), we see that our generalized gradient step (ie. the solution to (10)) is

$$\hat{\beta} = \left(1 - \frac{t(1-\alpha)\lambda}{||S(\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0),\, t\alpha\lambda)||_2}\right)_+ S(\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0),\, t\alpha\lambda). \tag{12}$$

If we iterate (12), and recenter each pass at $(\beta_0)_{\text{new}} = (\hat{\beta})_{\text{old}}$, then we will converge on the optimal solution for $\hat{\beta}^{(k)}$ given fixed values of the other coefficient vectors. If we apply this per block, and cyclically iterate through the blocks we will converge on the overall optimum. For ease of notation in the future we let $U(\beta_0, t)$ denote our update formula

$$U(\beta_0, t) = \left(1 - \frac{t(1-\alpha)\lambda}{||S(\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0),\, t\alpha\lambda)||_2}\right)_+ S(\beta_0 - t\nabla\ell(r_{(-k)}, \beta_0),\, t\alpha\lambda). \tag{13}$$

Note that in our case (linear regression)

$$\nabla\ell(r_{(-k)}, \beta_0) = -X^{(k)\top} r_{(-k)}/n.$$

3.2. **Algorithm Overview.** This algorithm is a sequence of nested loops:

(1) (Outer loop) Cyclically iterate through the groups; at each group (k) execute step 2
(2) Check if the group's coefficients are identically 0, by seeing if they obey

$$\left|\left|\mathrm{S}\left(X^{(k)\top} r_{(-k)},\, \alpha\lambda\right)\right|\right|_2 \leq (1-\alpha)\lambda.$$

If not, within the group apply step 3
(3) (Inner loop) Until convergence iterate:
   (a) update the center by $\theta \leftarrow \hat{\beta}^{(k)}$
   (b) update $\hat{\beta}^{(k)}$ from Eq (13), by

$$\hat{\beta}^{(k)} \leftarrow U(\theta, t)$$

This is the basic idea behind our algorithm. Meier et al. [2008] have a similar approach to fit the group lasso for generalized linear models. For a convergence threshold of ϵ, in the worst-case scenario within each group this algorithm requires $O(1/\epsilon)$ steps to converge. However, recent work in first order methods have shown vast improvements to gradient descent by a simple modification. As seen in Nesterov [2007] we can improve this class of algorithm to $O(1/\sqrt{\epsilon})$, by including a momentum term (known as accelerated generalized gradient descent). In practice as well, we have seen significant empirical improvement by including momentum in our gradient updates. We have also included step size optimization, which we have found important as often the lipschitz constant for a problem of interest is unknown. The actual algorithm that we employ changes the inner loop to the following:

(Inner loop) Start with $\beta^{(k,l)} = \theta^{(k,l)} = \beta_0^{(k)}$, step size $t = 1$, and counter $l = 1$. Repeat the following until convergence

(1) Update gradient g by $g = \nabla \ell \left(r_{(-k)}, \beta^{(k,l)} \right)$
(2) Optimize step size by iterating $t = 0.8 * t$ until

$$\ell \left(U \left(\beta^{(k,l)}, t \right) \right) \leq \ell \left(\beta^{(k,l)} \right) + g^\top \Delta_{(l,t)} + \frac{1}{2t} \left|\left| \Delta_{(l,t)} \right|\right|_2^2$$

(3) Update $\theta^{(k,l)}$ by

$$\theta^{(k,l+1)} \leftarrow U \left(\beta^{(k,l)}, t \right) \tag{14}$$

(4) Update the center via a Nesterov step by

$$\beta^{(k,l+1)} \leftarrow \theta^{(k,l)} + \frac{l}{l+3} \left(\theta^{(k,l+1)} - \theta^{(k,l)} \right) \tag{15}$$

(5) Set $l = l + 1$.

Where $\Delta_{(l,t)}$ is the change between our old solution and new solution

$$\Delta_{(l,t)} = U \left(\beta^{(k,l)}, t \right) - \beta^{(k,l)}$$

Our choice of 0.8 in step 2 was somewhat arbitrary; any value in $(0, 1)$ will work. This is very similar to the basic generalized gradient algorithm — the major differences are steps 2 and 4. In 2, we search for a t such that in our direction of descent, the majorization scheme still holds. In 4 we apply Nesterov-style momentum updates — this allows us to leverage some higher order information while only calculating gradients. While these momentum updates are unintuitive they have shown great theoretical and practical speedup in a large class of problems.

3.3. **Pathwise solution.** Usually, we will be interested in models for more than one amount of regularization. One could solve over a 2 dimensional grid of these α and λ values, however we found this to be computationally impractical, and to do a poor job of model selection. Instead, we fix the mixing parameter α and compute solutions for a path of λ values (as λ regulates the degree of sparsity). We begin the path with λ sufficiently large to set $\hat{\beta} = 0$, and decrease λ until we are near the unregularized solution. By using the previous solution as the start position for our algorithm at the next λ-value along the path, we make this procedure efficient for finding a pathwise solution. Notice that in Eq 5 if

$$||S\left(X^{(l)\top} y/n, \lambda\alpha\right)||_2 < \sqrt{p_l}(1-\alpha)\lambda$$

for all l, then $\beta = 0$ minimizes the objective. We can leverage the fact that $||S\left(X^{(l)\top} y/n, \lambda\alpha\right)||_2^2 - p_l(1-\alpha)^2\lambda^2$ is piecewise quadratic to find the smallest λ_l for each group that sets that group's coefficients to 0. Thus, we begin our path with

$$\lambda^{\max} = \max_i \lambda_i$$

This is the exact value at which the first coefficient enters the model. We choose $\lambda^{\min}$ to be some small fraction of $\lambda^{\max}$ (default value is 0.05 in our implementation) and log-linearly interpolate between these two for other values of λ on this path. While we do not have a theoretically optimal value for α we have found that using $\alpha \sim 0.95$ works well in problems where we expect overall sparsity but would also like to encourage grouping.

3.4. **Simple Extensions.** We can also use this algorithm to fit either the lasso or group lasso penalty: setting $\alpha = 1$ or $\alpha = 0$. For the group lasso the only change is to remove the soft thresholding in update (13) and get

$$U\left(\beta_0, t\right) = \left(1 - \frac{t(1-\alpha)\lambda}{||\beta_0 + t\nabla\ell(r_{(-i)}, \beta_0)||_2}\right)_+ ||\beta_0 + t\nabla\ell(r_{(-i)}, \beta_0)||_2.$$

For the lasso penalty, the algorithm changes a bit more. There is no longer any grouping, so there is no need for an outer group loop. Our update becomes

$$U\left(\beta_0, t\right) = S(\beta_0 + t\nabla\ell(y, \beta_0),\, t\lambda)$$

which we iterate, updating β_0 at each step. Without backtracking, this is just the NESTA algorithm in Lagrange form as described in Becker et al. [2009].

4. Extensions to other models

With little effort we can extend the sparse-group penalty to other models. If the likelihood function, $L(\beta)$, for the model of interest is log-concave then for the sparse-group lasso we minimize

$$\ell(\beta) + (1-\alpha)\lambda \sum_{l=1}^{m} \sqrt{p_l} \left|\left|\beta^{(l)}\right|\right|_2 + \alpha\lambda \left|\left|\beta\right|\right|_1$$

where $\ell(\beta) = -1/n \log\left(L(\beta)\right)$. Two commonly used cases, which we include in our implementation, are logistic regression and the Cox model for survival data.

For logistic regression we have y, an n-vector of binary responses, and X, an n by p covariate matrix divided into m groups, $X^{(1)}, \ldots, X^{(m)}$. In this case the sparse-group lasso takes the form

$$\hat{\beta} = \text{argmin}_\beta \frac{1}{n} \left[\left(\sum_{i=1}^{n} \log\left(1 + \exp\left(x_i^\top \beta\right)\right) + y_i x_i^\top \beta \right) \right] + (1-\alpha)\lambda \sum_{l=1}^{m} \sqrt{p_l} \left|\left|\beta^{(l)}\right|\right|_2 + \alpha\lambda \left|\left|\beta\right|\right|_1$$

For Cox regression our data is a covariate matrix, X (again with submatrices by group), an n-vector y corresponding to failure/censoring times and an n-vector δ indicating failure or censoring for each observation ($\delta_i = 1$ if observation i failed, while $\delta_i = 0$ if censored). Here the sparse-group lasso corresponds to

$$\hat{\beta} = \text{argmin}_\beta \frac{1}{n} \left[\log\left(\sum_{i \in D} \left(\sum_{j \in R_i} \exp\left(x_j^\top \beta\right) - x_i^\top \beta \right) \right) \right] + (1-\alpha)\lambda \sum_{l=1}^{m} \sqrt{p_l} \left|\left|\beta^{(l)}\right|\right|_2 + \alpha\lambda \left|\left|\beta\right|\right|_1$$

where D is the set of failure indices, R_i is the set of indices, j, with $y_j \geq y_i$ (those still at risk at failure time i).

4.1. **Fitting extensions.**

Fitting the model in these cases is straightforward. As before we use blockwise descent. Within each block our algorithm is nearly identical to the squared error case. While before we had

$$\ell(r_{(-k)}, \beta) = \frac{1}{2} \left|\left| r_{(-k)} - X^{(k)}\beta \right|\right|_2^2,$$

that form is only applicable with squared error loss. We define $\ell_k(\beta^{(-k)}, \beta^{(k)})$ to be our unpenalized loss function, $\ell(\beta)$, considered as a function of *only* $\beta^{(k)}$, with the rest of the coefficients, $\beta^{(-k)}$, fixed. In the case of square error loss, this is exactly $\ell(r_{(-k)}, \beta^{(k)})$. From here, we can use the algorithm in Section 3 only replacing every instance of $\ell(r_{(-k)}, \beta)$ by $\ell_k(\beta^{(-k)}, \beta^{(k)})$. We would like to note that although the algorithm

employed is straightforward, due to the curvature of these losses, in some cases our algorithm scales poorly (eg. Cox regression).

5. Simulated data

We compare the regular lasso to the sparse-group lasso for variable selection on simulated data. We simulated our covariate matrix X with different numbers of covariates, observations, and groups. The columns of X were iid. gaussian, and the response, y was constructed as

$$y = \sum_{l=1}^{g} X^{(l)}\beta^{(l)} + \sigma\epsilon \tag{16}$$

where $\epsilon \sim N(0, I)$, $\beta^{(l)} = (1, 2, \ldots, 5, 0, \ldots, 0)$ for $l = 1, \ldots, g$, and σ set so that the signal to noise ratio was 2. The number of generative groups, g varied from 1 to 3 changing the amount of the sparsity.

We chose penalty parameters for both the lasso and sparse-group lasso so that the number of nonzero coefficients chosen in the fits matched the true number of nonzero coefficients in the generative model (16) (5, 10, or 15 corresponding to $g = 1, 2, 3$). We then compared the proportion of correctly identified covariates averaged over 10 trials. Referring to Table 1, we can see that the sparse-group lasso improves performance in almost all scenarios. The two scenarios where the sparse-group lasso is slightly outperformed is unsurprising as there are few groups ($m = 10$) and each group has more covariates than observations ($n = 60$, $p = 150$), so we gain little by modeling sparsity of groups.

5.1. **Timings.**

We also timed our algorithm on simulated data for linear, logistic, and Cox regression. Our linear data was simulated as in section 5. To simulate binary responses, we applied a logit transformation to a scaling of our linear responses

$$p_i = \frac{\exp(5y_i)}{1 + \exp(5y_i)}$$

and simulated bernoulli random variables with these probabilities. For Cox regression, we set survival/censoring time for observation i to be $\exp(y_i)$, and simulated our indicators death/censoring independently with equal probability of censoring and death (ber(0.5)). We used the same covariate matrix for each 3 regression types. For the smaller data sets ($p = 1500$, and $p = 2000$) we used $\lambda_{min} = 0.2\lambda_{max}$. For the larger problems, traversing this far along the regularization path was unnecessary (the solution with minimal cross-validated error was reached much

	Number of Groups in Generative Model		
	1 group	2 groups	3 groups
	$n = 60$, $p = 1500$, $m = 10$		
SGL	0.72	0.36	0.28
Lasso	0.60	0.38	0.31
	$n = 70$, $p = 2000$, $m = 200$		
SGL	0.68	0.44	0.31
Lasso	0.54	0.30	0.26
	$n = 150$, $p = 10000$, $m = 100$		
SGL	0.77	0.72	0.52
Lasso	0.76	0.62	0.43
	$n = 200$, $p = 20000$, $m = 400$		
SGL	0.92	0.78	0.68
Lasso	0.82	0.68	0.52

TABLE 1. Proportions of correct nonzero coefficient identifications for standardized and unstandardized Group Lasso out of 10 simulated data sets.

earlier), and less sparse points in the regularization path are inefficient to solve, so we used $\lambda_{min} = 0.5\lambda_{max}$. All timings were carried out on an Intel Xeon 3.33 GHz processor

Referring to Table 2, we see that our algorithm scales reasonably efficiently. In the linear case, problems can be solved in a matter of seconds. Logistic and Cox regression run more slowly, however they still run within minutes on larger datasets. One noteworthy point is that smaller group sizes allow our algorithm to make better use of active sets, and this is reflected in runtimes between the 200 and 10 group cases. Also, one may find the shorter run-times for larger problems

	Number of Groups in Generative Model		
	1 group	2 groups	3 groups
	$n = 60$, $p = 1500$, $m = 10$		
linear	12.8	37.2	37.6
logit	42.9	47.9	49.8
cox	54.5	55.4	57.8
	$n = 70$, $p = 2000$, $m = 200$		
linear	2.27	6.39	10.1
logit	18.1	28.7	28.2
cox	34.1	39.3	36.2
	$n = 150$, $p = 10000$, $m = 100$		
linear	7.3	14.3	23.3
logit	17.4	91.1	121
cox	62.13	227.3	217.5
	$n = 200$, $p = 20000$, $m = 400$		
linear	10.5	14.4	19.9
logit	13.1	55.6	110.5
cox	69.2	219.5	273.8

TABLE 2. Time in seconds to solve for a path of 20 λ-values averaged over 10 simulated data sets.

confusing — these problems have smaller group sizes, and because we need not run too far along the regularization path, the group-size and group sparsity have a greater effect on runtime than overall problem size.

6. Discussion

We have proposed and given insight into a method for modeling groupwise and within group sparsity in regression. We have extended

this model to other likelihoods. We have shown the efficacy of this method on simulated data, and given an efficient algorithm to fit this model. An `R` implementation of this algorithm is available on request, and will soon be uploaded to `CRAN`.

7. Supplemental Materials

R Files: The `R` library for running our fitting code, and the scripts for running all the timing and accuracy simulations in the manuscript are available in the supplemental materials online.

References

S. Becker, J. Bobin, and E. Candes. NESTA: A fast and accurate first-order method for sparse recovery. *Arxiv preprint arXiv*, 904, 2009.

R. Foygel and M. Drton. Exact block-wise optimization in group lasso and sparse group lasso for linear regression. *Arxiv preprint arXiv:1010.3320*, 2010.

J. Friedman, T. Hastie, and R. Tibshirani. A note on the group lasso and sparse group lasso. *arViV:1001.0736v1*.

J. Friedman, T. Hastie, and R. Tibshirani. Applications of the lasso and grouped lasso to the estimation of sparse graphical models. submitted, 2009.

L. Meier, S. van de Geer, and P. Bühlmann. The group lasso for logistic regression. *Journal of the Royal Statistical Society B*, 70:53–71, 2008.

Y. Nesterov. *Gradient methods for minimizing composite objective function*. CORE, 2007.

A. Puig, A. Wiesel, and A. Hero. A multidimensional shrinkage-thresholding operator," statistical signal processing. In *SSP '09. IEEE/SP 15th Workshop on Statistical Signal Processing*, pages 113–116, 2009.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society B*, 58:267–288, 1996.

M. Yuan and Y. Lin. Model selection and estimation in regression with grouped variables. *Journal of the Royal Statistical Society, Series B*, 68(1):49–67, 2007.

H. Zou and T. Hastie. Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society B*, 67(2):301–320, 2005.