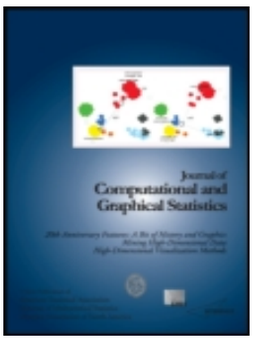

Taylor & Francis
Taylor & Francis Group

Journal of Computational and Graphical Statistics

ISSN: 1061-8600 (Print) 1537-2715 (Online) Journal homepage: http://www.tandfonline.com/loi/ucgs20

A Sparse-Group Lasso

Noah Simon , Jerome Friedman , Trevor Hastie & Robert Tibshirani

To cite this article: Noah Simon , Jerome Friedman , Trevor Hastie & Robert Tibshirani (2013) A Sparse-Group Lasso, Journal of Computational and Graphical Statistics, 22:2, 231-245, DOI: 10.1080/10618600.2012.681250

To link to this article: https://doi.org/10.1080/10618600.2012.681250

Supplementary materials for this article are available online.
Please go to www.tandfonline.com/r/JCGS

A Sparse-Group Lasso

Noah SIMON, Jerome FRIEDMAN, Trevor HASTIE, and Robert TIBSHIRANI

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 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. This article has online supplementary material.

Key Words: Model; Nesterov; Penalize; Regression; Regularize.

1. INTRODUCTION

Consider the usual linear regression framework. Our data consist 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 that uses only a few of the groups. Yuan and Lin (2007) proposed the group-lasso criterion for this problem; the problem is

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

Noah Simon, Department of Statistics, Stanford University (E-mail: *nsimon@stanford.edu*). Jerome Friedman, Department of Statistics, Stanford University (E-mail: *jhf@stanford.edu*). Trevor Hastie, Department of Statistics, Department of Health Research and Policy, Stanford University (E-mail: *hastie@stanford.edu*). Robert Tibshirani, Department of Statistics, Department of Health Research and Policy, Stanford University (E-mail: *tibs@stanford.edu*).

Journal of Computational and Graphical Statistics, Volume 22, Number 2, Pages 231–245
DOI: 10.1080/10618600.2012.681250

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). Before we move on, we would like to define consistent terminology for our two types of sparsity: we use "groupwise sparsity" to refer to the number of groups with at least one nonzero coefficient, and "within group sparsity" to refer to the number of nonzero coefficients within each nonzero group. Occasionally, we will also use the term "overall sparsity" to refer to the total number of nonzero coefficients regardless of grouping.

In this article we discuss properties of this criterion, first proposed in our earlier unpublished note (Friedman et al. 2010). We discuss using this idea for 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 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` (see online supplementary material, posted on the journal web site). This article is the continuation of Friedman et al. (2010), a brief note on the criterion.

This criterion was also discussed in the article by Zhou et al. (2010). They applied it to SNP data for linear and logistic regression with an emphasis on variable selection and found that it performed well.

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 discuss when we might expect our model to outperform the lasso and group lasso, and give some real data examples. In Section 6, 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

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

For the rest of the article, 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 Equation (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

$$\frac{1}{n}X^{(k)\top}\left(y-\sum_{l=1}^{m}X^{(l)}\hat{\beta}^{(l)}\right)=(1-\alpha)\lambda u+\alpha\lambda v,$$

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

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

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

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

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

where $r_{(-k)}$ 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 with $S(\cdot)$ the coordinate-wise soft thresholding operator:

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

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

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

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 that 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

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

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.

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

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

These are elastic net type conditions as mentioned in the article by Friedman, Hastie, and Tibshirani (2009). Unlike the usual elastic net, the proportional shrinkage here is a function of the optimal solution, $\lambda_{\text{net},2} = (1-\alpha)\lambda/||\hat{\beta}^{(k)}||_2$. Equation (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, Wiesel, and Hero (2009) and Foygel and Drton (2010) fitted the group lasso and sparse-group lasso, respectively, by explicitly solving for $||\hat{\beta}^{(k)}||_2$ and applying Equation (7) in a cyclic fashion for each group with all other groups fixed. 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 such that the quadratic term dominates the Hessian of our loss; note that 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 Equation (9) majorizes the objective in Equation (8).

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

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 Equation (11), we see that our generalized gradient step (i.e., the solution to Equation (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 Equation (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 iterating 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 updated 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

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

   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 Equation (13), by

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

This is the basic idea behind our algorithm. Meier, van de Geer, and Bühlmann (2008) had 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 $\mathrm{O}(1/\epsilon)$ steps to converge. However, recent work in first-order methods have shown vast improvements to gradient descent by a simple modification. As seen in the literature by Nesterov (2007), we can improve this class of algorithm to $\mathrm{O}(1/\sqrt{\epsilon})$, 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 $g = \nabla \ell(r_{(-k)}, \beta^{(k,l)})$;
2. optimize step size by iterating $t = 0.8 * t$ until

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

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

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

4. update the center via a Nesterov step by

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

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 Step 2, we search for a t such that in our direction of descent the majorization scheme still holds. In Step 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 two-dimensional grid of 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 Equation (5), if

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

for all l, then $\beta = 0$ minimizes the objective. We can leverage the fact that for a fixed α, $||S(X^{(l)\top} y/n, \lambda\alpha)||_2^2 - p_l(1-\alpha)^2\lambda^2$ is piecewise quadratic in λ 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 0.1 in our implementation) and log-linearly interpolate between these two for other values of λ on this path. We do not have a theoretically optimal value for α—the optimal value would need to be a function of the number of covariates and group sizes among other things. In practice, for problems where we expect strong overall sparsity and would like to encourage grouping, we have used $\alpha = 0.95$ with reasonable success (this was used in our simulated data in Section 6). In contrast, if we expect strong group-wise sparsity, but only mild sparsity within group, we have used $\alpha = 0.05$ (an example of this is given in Section 5). That said, different problems will possibly be better served by different values of α and in practice some exploration may be needed.

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 updated Equation (13) and get

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

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

$$U(\beta_0, t) = S(\beta_0 - t\nabla\ell(y, \beta_0),\ t\lambda),$$

which we iterate, updating β_0 at each step. Without backtracking, this is just the NESTA algorithm in Lagrange form as described in the article by Becker, Bobin, and Candes (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

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

where $\ell(\beta) = -1/n\log(L(\beta))$. 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

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

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

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

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. Previously, 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 (e.g., Cox regression).

4.2 OVERLAP GROUP LASSO

The sparse-group lasso can also be considered as a special case of a group-lasso model that allows overlap in groups (in this case many groups would be size 1). In the more general overlap case one, may see strange behavior—if a variable in the overlap of many groups is included in a model then all of those groups will need to be "active." Jacob, Obozinski, and Vert (2009) gave a very nice fix for this issue by slightly reformulating the problem, but this more general framework is beyond the scope of our article.

5. APPLICATIONS

In this section we discuss when one might expect good performance from the sparse-group lasso, and when another tool might be preferable. One common statistical scenario is regression with categorical predictors. For predictors with few levels, it is reasonable to use the group lasso—sparsity within group is unnecessary as groups are small. As the number of levels per predictor rises, it becomes more likely that even for predictors that we include, many of the levels may not be informative. The sparse-group lasso will take this into account, setting the coefficients for many levels equal to 0 even in nonzero groups. At the other extreme, few predictors each with many levels, groupwise sparsity often proves unhelpful and one may see the best performance with the lasso (e.g., if there are only five groups and one is active in the true model, this is still 20% of groups active).

Along similar lines, often we run regression in a setting where the predictors have a natural grouping. We mentioned gene pathways before and will expand on it here. In many (if not all) genetic conditions, genes do not function (or fail to function) independently. If numbers of genes in a given pathway all seem moderately successful at predicting outcome, we would like to up-weight this evidence over similarly predictive genes in different pathways. However, we also do not believe that every gene in an active pathway is

necessarily indicated in the genetic condition. The sparse-group lasso is potentially useful for this scenario—it finds pathways of interest and, from them, selects driving genes. Furthermore, it shrinks the estimated effects of driving genes within a group toward one another.

To further investigate this, we have compared the sparse-group lasso to the lasso and group lasso on two real data examples with gene expression data. Our first dataset was the colitis data by Burczynski et al. (2006). There were a total of 127 patients, 85 with colitis (59 Crohn's patients + 26 ulcerative colitis patients) and 42 healthy controls. Each patient had expression data for 22, 283 genes run on an affymetrix U133A microarray. These genes were grouped into "genesets" using cytogenetic position data (the $C1$ set from Subramanian et al. 2005). Of the original 22, 283 genes, only 8298 of these were found in the $C1$ geneset—the others were removed from the analysis. The $C1$ set contains 277 cytogenetic bands, each averaging about 30 genes (from out dataset). We chose 50 observations at random and used these to fit our models. We used the remaining 77 observations as a test set.

Because there were a large number of small pathways, we chose $\alpha = 0.05$ for the sparse-group lasso model. Each of the three models was fit for a path of 100 λ-values with $\lambda_{\min} = 0.01\lambda_{\max}$ (this value was chosen because the peak in validation accuracy occurred at the end of the path for $\lambda_{\min} = 0.1\lambda_{\max}$).

In Figure 1, we see that the lasso performed slightly better than the group lasso and sparse-group lasso with a 90% correct classification rate at its peak to 87% for the sparse-group lasso and 84% for the group lasso. If we look into the solution slightly more we see that, at its peak, the lasso chose to include 19 genes from totally different cytogenetic bands, whereas the sparse-group lasso included 43 genes from only eight cytogenetic bands, and the group lasso included all 36 genes from seven bands. In this example, the group lasso

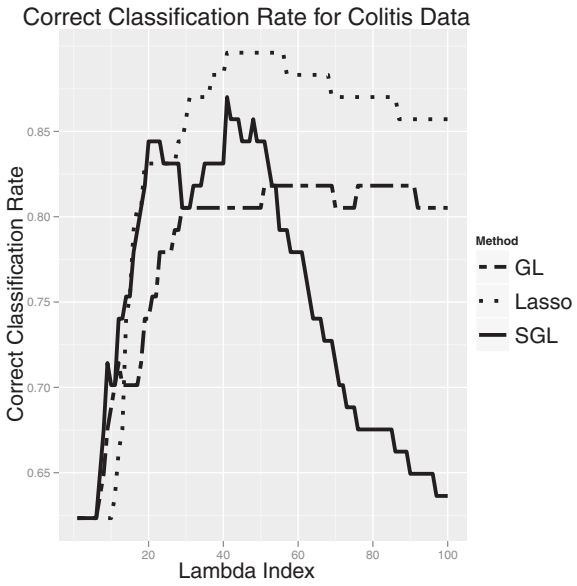

Figure 1. Classification accuracy on 77 test samples for colitis data. All models were fit for a path of 100 λ-values with $\lambda_{\min} = 0.1\lambda_{\max}$. For the SGL, $\alpha = 0.05$ was used.

and sparse-group lasso chose nearly the same predictors (the sparse-group lasso included an additional group). All of these bands were very small (the largest had 11 genes, 4.5 was the median), and the sparse-group lasso actually did not employ any within-group sparsity. From our results, we can see that the sparse-group lasso (with our choice of gene sets) was not ideal for this problem.

One might question our choice of the positional bands rather than another collection of gene sets. We chose the $C1$ collection because it seemed reasonable and had no overlapping groups. In general, scientists with domain-specific knowledge would likely do better choosing a domain-specific collection (e.g., using a colitis-associated collection for the colitis data).

The second dataset we used for comparison was the breast cancer data by Ma et al. (2004). This dataset contains gene expression values from 60 patients with estrogen positive breast cancer. The patients were treated with tamoxifen for 5 years and classified according to whether cancer recurred (there were 28 recurrences). Gene expression values were run on a GPL1223: Arcturus 22k human oligonucleotide microarray. Unfortunately, there was significant missing data. As a first pass, all genes with more than 50% missing data were removed. Other missing values were imputed by simple mean imputation. This left us with $12,071$ of our $22,575$ original genes. We again grouped genes together by cytogenetic position data, removing genes that were not recorded in the GSEA C1 dataset. Our final design matrix had 4989 genes in 270 pathways (an average of ~ 18.5 genes per pathway). Thirty patients were chosen at random and used to build the three models. The remaining 30 were used to test their accuracies.

We again used $\alpha = 0.05$ for the sparse-group lasso. Each of the three models was fit for a path of 100 λ-values with $\lambda_{\min} = 0.1\lambda_{\max}$.

Referring to Figure 2, we see that in this example the sparse-group lasso outperforms the lasso and group lasso. The sparse-group lasso reaches 70% classification accuracy (though this is a narrow peak, so may be slightly biased high), while the group lasso peaks at 60% and the lasso comes in last at 53% accuracy. At its optimum the sparse-group lasso includes 54 genes from 11 bands, while the group lasso selects all 74 genes from 15 bands (again, largely smaller bands for the group lasso), and the lasso selects three genes all from separate bands. This example really highlights the advantage of the sparse-group lasso—it allows us to use group information, but does not force us to use entire groups.

These two examples highlight two different possibilities for the sparse-group lasso. In the cancer data, the addition of group information is critical for classification, and the grouping may help give insight into the biological mechanisms. In the colitis data, the group "information" largely just increases model variance. The sparse-group lasso is certainly not perfect for every scenario with grouped data, but as evidenced in the cancer data it can sometimes be helpful.

6. SIMULATED DATA

In the previous section, we compared the predictive accuracy of the lasso and sparse-group lasso on real data. One might also be interested in its accuracy as a variable selection tool—in this section, we compare the regular lasso to the sparse-group lasso for variable

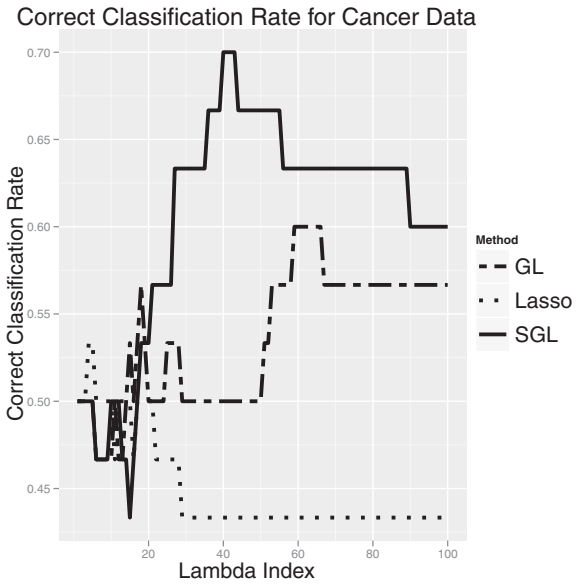

Figure 2. Classification accuracy on 30 test samples for cancer data. Both models were fit for a path of 100 λ-values with $\lambda_{\min} = 0.1\lambda_{\max}$. For the SGL, $\alpha = 0.05$ was used.

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 one to three changing the amount of the sparsity.

We chose penalty parameters for both the lasso and sparse-group lasso (with $\alpha = 0.95$) so that the number of nonzero coefficients chosen in the fits matched the true number of nonzero coefficients in the generative model in Equation (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.

We would like to note that in some trials we were unable to make the sparse-group lasso select exactly the true number of nonzero coefficients (due to the grouping effects). In these cases, we allowed the sparse-group lasso to select extra variables (as few as it could manage); however, when calculating the proportion of correct nonzero coefficient identifications, we used the total number of variables selected in our denominator, for example, if the sparse-group lasso selected seven variables in the five true variable case, it

Table 1. Proportions of correct nonzero coefficient identifications for standardized and unstandardized group lasso out of 10 simulated datasets

	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

would be unable to get a proportion better than $5/7 = 0.71$. While not ideal, we find no reason to believe that this would bias our results in favor of the sparse-group lasso.

6.1 TIMINGS

We also timed our algorithm on simulated data for linear, logistic, and Cox regression. Our linear data was simulated as in Section 6. 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 three regression types.

We found that the unregularized end of the regularization path required by far the most time to solve. To illustrate this, we ran two sets of simulations. For the first set we use $\lambda_{\min} = 0.1\lambda_{\max}$, running relatively far along the path. For the second set we ran a much shorter path with $\lambda_{\min} = 0.6\lambda_{\max}$. For some problems it may be necessary to solve for $\lambda_{\min}$ small. However, these solutions have many nonzero variables. As such in large p, small n problems, these unregularized solutions generally have very poor prediction accuracy as they tend to include many noise variables. In these situations, solving far into the regularization path may often be unnecessary.

Our implementation of the sparse-group lasso is called from R, but much of the optimization code is written in C++ and compiled as a shared R/C++ library. All timings were carried out on an Intel Xeon 3.33 GHz processor.

Referring to Table 2, we see that while, in some cases, our algorithm scales somewhat poorly, it can still solve fairly large problems with times on the order of minutes. One noteworthy point is that smaller group sizes allow our algorithm to make better use of active sets, and this is reflected in the runtime differences between the 200 and 10 group

Table 2. Time in seconds to solve for the "short" and "long" paths of 20 λ-values averaged over 10 simulated datasets, where $\lambda_{\min} = 0.6\lambda_{\max}$ for the short path and $\lambda_{\min} = 0.1\lambda_{\max}$ for the long path

	Number of groups in generative model					
	Short path			Long path		
	1 group	2 groups	3 groups	1 group	2 groups	3 groups
			$n = 150,\ p = 1500,\ m = 10$			
linear	1.788	7.052	8.278	30.86	65.13	66.29
logit	5.954	7.055	8.72	58.21	63.14	65.12
cox	7.592	10.09	8.453	72.87	76.31	77.17
			$n = 200,\ p = 2000,\ m = 200$			
linear	0.501	0.8164	1.345	8.422	14.18	18.35
logit	1.594	3.121	2.676	32.48	42.38	43.65
cox	2.482	4.084	3.315	55.31	58	53.41
			$n = 150,\ p = 10{,}000,\ m = 100$			
linear	6.566	11.46	15.15	140.4	269.6	322
logit	8.017	48.01	60.94	424.1	554.9	595.6
cox	21.61	117.3	117.2	635.8	793.4	789.2
			$n = 200,\ p = 20{,}000,\ m = 400$			
linear	9.743	13.07	15.75	153.3	272.3	401.3
logit	10.73	23.6	50.87	600.8	815.7	965.4
cox	23.86	91.2	128.2	1324	1473	1578

cases. Also, as we run further into the regularization path, more groups become active. This is the main reason our solutions for $\lambda_{\min} = 0.1$ are much slower than for $\lambda_{\min} = 0.6$ even though the two paths solve for the same number of λ-values.

7. 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 real and simulated data, and given an algorithm to fit this model. An R implementation of this algorithm is available on request, and will soon be uploaded to CRAN.

SUPPLEMENTARY MATERIALS

R-package for SGL: R-package "SGL" containing the code used to fit all the spare-group lasso models. (SGL_1.0.tar.gz, GNU zipped tar file)

Test Code: All the R script files used for simulations and real data calculations run in the article. (testCode.tar.gz, GNU zipped tar file)

[Received May 2011. Revised February 2012.]

REFERENCES

Becker, S., Bobin, J., and Candes, E. (2009), "NESTA: A Fast and Accurate First-Order Method for Sparse Recovery," *arXiv:0904:3367v1*. [238]

Burczynski, M., Peterson, R., Twine, N., Zuberek, K., Brodeur, B., Casciotti, L., Maganti, V., Reddy, P., Strahs, A., Immermann, F., Spinelli, W., Schwertschlag, U., Slager, A. M., Cotreau, M. M., and Dorner, A. J. (2006), "Molecular Classification of Crohn's Disease and Ulcerative Colitis Patients Using Transcriptional Profiles in Peripheral Blood Mononuclear Cells," *Journal of Molecular Diagnostics*, 8, 51. [240]

Foygel, R., and Drton, M. (2010), "Exact Block-Wise Optimization in Group Lasso and Sparse Group Lasso for Linear Regression," *arXiv:1010.3320v2*. [234]

Friedman, J., Hastie, T., and Tibshirani, R. (2010), "Regularization Paths for Generalized Linear Models via Coordinate Descent," *Journal of Statistical Software*, 33, 1. [234]

——— (2010), "A Note on the Group Lasso and Sparse Group Lasso," *arXiv:1001.0736v1*. [232]

Jacob, L., Obozinski, G., and Vert, J. (2009), "Group Lasso With Overlap and Graph Lasso," in ACM *Proceedings of the 26th Annual International Conference on Machine Learning*, pp. 433–440. [239]

Ma, X., Wang, Z., Ryan, P., Isakoff, S., Barmettler, A., Fuller, A., Muir, B., Mohapatra, G., Salunga, R., Tuggle, J., Tran, Y., Tran, D., Tassin, A., Amon, P., Wang, W., Wang, W., Enright, E., Stecker, K., Estepa-Sabal, E., Smith, B., Younger, J., Balis, U., Michaelson, J., Bhan, A., Habin, K., Baer, T. M., Brugge, J., Haber, D. A., Erlander, M. G., and Sgroi, D. C. (2004), "A Two-Gene Expression Ratio Predicts Clinical Outcome in Breast Cancer Patients Treated With Tamoxifen," *Cancer Cell*, 5, 607–616. [241]

Meier, L., van de Geer, S., and Bühlmann, P. (2008), "The Group Lasso for Logistic Regression," *Journal of the Royal Statistical Society,* Series B, 70, 53–71. [236]

Nesterov, Y. (2007), *Gradient Methods for Minimizing Composite Objective Function*, CORE. [236]

Puig, A., Wiesel, A., and Hero, A. (2009), "A Multidimensional Shrinkage-Thresholding Operator, Statistical Signal Processing," in *SSP'09, IEEE/SP 15th Workshop on Statistical Signal Processing*, pp. 113–116. [234]

Subramanian, A., Tamayo, P., Mootha, V., Mukherjee, S., Ebert, B., Gillette, M., Paulovich, A., Pomeroy, S., Golub, T., Lander, E., and Mesirov, J. P. (2005), "Gene Set Enrichment Analysis: A Knowledge-Based Approach for Interpreting Genome-Wide Expression Profiles," *Proceedings of the National Academy of Sciences of the United States of America*, 102, 15545–15550. [240]

Tibshirani, R. (1996), "Regression Shrinkage and Selection Via the Lasso," *Journal of the Royal Statistical Society,* Series B, 58, 267–288. [231]

Yuan, M., and Lin, Y. (2007), "Model Selection and Estimation in Regression With Grouped Variables," *Journal of the Royal Statistical Society,* Series B, 68, 49–67. [231]

Zhou, H., Sehl, M., Sinsheimer, J., and Lange, K. (2010), "Association Screening of Common and Rare Genetic Variants by Penalized Regression," *Bioinformatics*, 26, 2375–2382. [232]

Zou, H., and Hastie, T. (2005), "Regularization and Variable Selection Via the Elastic Net," *Journal of the Royal Statistical Society,* Series B, 67, 301–320. [233]