Journal of Statistical Software

MMMMMM YYYY, Volume VV, Issue II. *http://www.jstatsoft.org/*

A Blockwise Descent Algorithm for Group-penalized Multiresponse and Multinomial Regression

Noah Simon
Stanford University

Jerome Friedman
Stanford University

Trevor Hastie
Stanford University

Abstract

In this paper we purpose a blockwise descent algorithm for group-penalized multiresponse regression. Using a quasi-newton framework we extend this to group-penalized multinomial regression. We give the first publicly available implementation for these in `R`, and compare the speed of this algorithm to a similar algorithm for standard ℓ_1-penalized multinomial regression on simulated data — we show that our implementation is roughly as fast and can solve gene-expression-sized problems in real time.

Keywords: multiresponse, multinomial, lasso, penalized, coordinate descent.

1. Introduction

Consider the usual linear regressions framework with y an n-vector of responses and X, an n by p matrix of covariates. Traditionally, problems involve $n < p$ and it is standard to estimate a regression line with least squares. In many recent applications (genomics and advertising, among others) we have $p \gg n$, and standard regression fails. In these cases, one is often interested in a solution involving only few covariates. Toward this end, Tibshirani (1996) proposed the Lasso: to find our regression coefficients by solving the regularized least squares problem

$$\hat{\beta} = \text{argmin}_\beta \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \|\beta\|_1$$

This method regularizes β by trading off "goodness of fit" for a reduction in "wildness of coefficients" — it also has the effect of giving a solution with few nonzero entries in β. This was generalized by Yuan and Lin (2007) to deal with grouped covariates; they propose to

solve

$$\hat{\beta} = \text{argmin}_\beta \frac{1}{2} \|y - X\beta\|_2^2 + \lambda \sum_k \left\|\beta_{I(k)}\right\|_2 \tag{1}$$

where the covariates are partitioned into disjoint groups and $I(k)$ denotes the indices of the kth group of covariates (ie. $\beta_{I(k)}$ indicates the sub-vectors of β corresponding to group k). This approach gives a solution with few non-zero groups.

Now, instead of usual linear regression, one might be interested in multiresponse regression — instead of an n-vector, Y is an $n \times M$ matrix, and β is a $p \times M$ matrix. In some cases one might believe that our response variables are related, in particular that they have roughly the same set of important explanatory variables, a subset of all predictor variables measured (ie. in each row of β either all of the elements are zero or all are non-zero.). A number of authors have similar suggestions for this problem (Obozinski, Tasker, and Jordan (2007), Argyriou, Evgeniou, and Pontil (2007), among others) which build on the group-lasso idea of Yuan and Lin (2007); to use a group-penalty on rows of β:

$$\hat{\beta} = \text{argmin}_\beta \frac{1}{2} \|Y - X\beta\|_2^2 + \lambda \sum_{k=1}^{p} \|\beta_{k\cdot}\|_2 \tag{2}$$

where $\beta_{k\cdot}$ refers to the kth row of β (likewise, we will, in the future, use $\beta_{\cdot m}$ to denote the mth column of β). For the remainder of this paper we will refer to (2) as the "multiresponse lasso".

Multinomial regression (via a generalized linear model) is also initimately related to multiresponse regression. In particular, common methods for finding the MLE in non-penalized multinomial regression (eg Newton-Raphson) reduce the problem to that of solving a series of weighted multiresponse regression problems. In multinomial regression, β is again an $n \times M$ matrix, where M is now the number of classes — the (k, m) entry gives the contribution of variable x_k to class m. More specifically, the probability of an observation with covariate vector x belonging to a class l is parametrized as

$$\text{P}\left(y = l|x\right) = \frac{\exp\left(x^\top \beta_{\cdot l}\right)}{\sum_{m \le M} \exp\left(x^\top \beta_{\cdot m}\right)}.$$

As in linear regression, one might want to fit a penalized version of this model. Standard practice has been to fit

$$\hat{\beta} = \text{argmin}_\beta -\ell\left(y, X\beta\right) + \lambda \|\beta\|_1 .$$

where ℓ is the multinomial log-likelihood. While this does give sparsity in β it may give a very different set of non-zero coefficients in each class. We instead discuss using

$$\hat{\beta} = \text{argmin}_\beta -\ell\left(y, X\beta\right) + \lambda \sum_{k=1}^{p} \|\beta_{k\cdot}\|_2 . \tag{3}$$

which we will refer to as the group-penalized multinomial lasso. Because each term in the penalty sum is non-differentiable only when all elements of the vector $\beta_{k\cdot}$ are 0, this formulation has the advantage of giving the same nonzero coefficients in each class. It was very recently proposed by Vincent and Hansen (2012). When the true underlying model has the

same (or similar) non-zero coefficient structure across classes, this model can improve prediction accuracy. Furthermore, this approach can lead to more interpretable model estimates.

One downside of this approach is that minimization of the criterion in (3) requires new tools. The criterion is convex so for small to moderate sized problems interior point methods can be used — however, in many applications there will be many features ($p > 10,000$) and possibly a large number of classes and/or observations. For the usual lasso, there are coordinate descent and first order algorithms that can scale to much larger problem sizes. These algorithms must be adjusted for the grouped multinomial lasso problem.

In this paper we discuss an efficient block coordinate descent algorithm to fit the group-penalized multiresponse lasso, and group-penalized multinomial lasso. The algorithm in this paper is the multiresponse analog to Friedman, Hastie, and Tibshirani (2010). In particular, we have incorporated this algorithm into `glmnet` a widely used `R` package for solving penalized regression problems.

2. Penalized Multiresponse Regression

We first consider our Gaussian objective, (2):

$$\min \frac{1}{2} \|Y - X\beta\|_F^2 + \lambda \sum_{k \le p} \|\beta_{k\cdot}\|_2$$

This can be minimized by blockwise coordinate descent (one row of β at a time). Consider a single $\beta_{k\cdot}$ with fixed $\beta_{j\cdot}$ for all $j \neq k$. Our objective becomes

$$\min \frac{1}{2} \|R_{-k} - X_{\cdot k}\beta_{k\cdot}\|_F^2 + \lambda \|\beta_{k\cdot}\|_2$$

where $X_{\cdot k}$ refers to the kth column of X, and $R_{-k} = Y - \sum_{j \neq k} X_{\cdot j}\beta_{j\cdot}$ is the partial residual. If we take a subderivative with respect $\beta_{k\cdot}$, then we get that $\hat{\beta}_{k\cdot}$ satisfies

$$\|X_{\cdot k}\|_2^2 \hat{\beta}_{k\cdot}^\top - X_{\cdot k}^\top R_{-k} + \lambda S(\hat{\beta}_{k\cdot}) = 0$$

where $S(\hat{\beta}_{k\cdot})$ is its sub-differential

$$S(a) \begin{cases} = \frac{a}{||a||_2}, & \text{if } a \neq 0 \\ \in \{u \text{ s.t. } ||u||_2 \le 1\}, & \text{if } a = 0 \end{cases}$$

From here, simple algebra gives us that

$$\hat{\beta}_{k\cdot} = \frac{1}{\|X_{\cdot k}\|_2^2} \left(1 - \frac{\lambda}{\left\|X_{\cdot k}^\top R_{-k}\right\|_2}\right)_+ X_{\cdot k}^\top R_{-k}$$

where $(a)_+ = \max(0, a)$. By cyclically applying these updates we can minimize our objective. The convergence guarantees of this style of algorithm are discussed in **?**.

Gaussian Algorithm

Now we have the following very simple algorithm. We can include all the other bells and whistles from `glmnet` as well

1. Initialize $\beta = \beta_0$, $R = Y - X\beta_0$
2. Iterate until convergence: for $k = 1, \ldots, p$
   (a) Update R_{-k} by
   $$R_{-k} = R + X_{\cdot k}\beta_{k\cdot}$$
   (b) Update $\beta_{k\cdot}$ by
   $$\beta_{k\cdot} \leftarrow \frac{1}{\|X_{\cdot k}\|_2^2}\left(1 - \frac{\lambda}{\left\|X_{\cdot k}^\top R_{-k}\right\|_2}\right)_+ X_{\cdot k}^\top R_{-k}$$
   (c) Update R by
   $$R = R_{-k} - X_{\cdot k}\beta_{k\cdot}$$

Note that in step (2*b*) if $\left\|X_{\cdot k}^\top R_{-k}\right\|_2 \leq \lambda$, the update is simply $\beta_{k\cdot} \leftarrow \mathbf{0}$.

If we would like to include intercept terms in the regression, we need only mean center the columns of X and Y before carrying out the algorithm (this is equivalent to a partial minimization with respect to the intercept term).

3. Extension to Multinomial Regression

We now extend this idea to the multinomial setting. Suppose we have M different classes. In this setting, Y is an $n \times M$ matrix of zeros and ones — the ith row has a single one corresponding to the class of observation i. In a multinomial generalized linear model one assumes that the probability of observation i coming from class m has the form

$$p_{i,m} = \frac{\exp(\eta_{i,m})}{\sum_{l \leq M} \exp(\eta_{i,l})}$$

with

$$\eta_{i,m} = X_{i\cdot}\beta_{\cdot m}.$$

This is the symmetric parametrization. From here we get the multinomial log-likelihood

$$\ell(\mathbf{p}) = \sum_{i=1}^{n}\sum_{m=1}^{M} y_{i,m}\log(p_{i,m})$$

which we can rewrite as

$$\begin{aligned}\ell(\boldsymbol{\eta}) &= \sum_{i=1}^{n}\sum_{m=1}^{M} y_{i,m}\left[\eta_{i,m} - \log\left(\sum_{l \leq M}\exp(\eta_{i,l})\right)\right] \\ &= \sum_{i=1}^{n}\left[\sum_{m=1}^{M} y_{i,m}\eta_{i,m} - \log\left(\sum_{l \leq M}\exp(\eta_{i,l})\right)\right]\end{aligned}$$

since $\sum_{m=1}^{M} y_{i,m} = 1$ for each i. Thus our final minimization problem is

$$\min -\sum_{i=1}^{n}\left[\sum_{m=1}^{M} y_{i,m}X_{i\cdot}\beta_{\cdot m} - \log\left(\sum_{l\leq M}\exp\left(X_{i\cdot}\beta_{\cdot l}\right)\right)\right] + \lambda\sum_{k\leq p}\|\beta_{k\cdot}\|_2 \tag{4}$$

3.1. Uniqueness

Before proceeding, we should note that, because we choose to use the symmetric version of the multinomial log-likelihood, without our penalty (or with $\lambda = 0$) the solution to this objective is never unique. If we add or subtract a constant to an entire row of β, the unpenalized objective is unchanged. To see this, consider replacing our estimate for the kth row $\hat{\beta}_{k\cdot}$ by $\hat{\beta}_{k\cdot} + \delta\mathbf{1}^\top$ (for some scalar δ). Then we have

$$\begin{aligned}
\hat{P}_\delta\left(y = l|x\right) &= \frac{\exp\left(x^\top\hat{\beta}_{\cdot l} + x_k\delta\right)}{\sum_{m\leq M}\exp\left(x^\top\hat{\beta}_{\cdot m} + x_k\delta\right)} \\
&= \frac{\exp\left(x_k\delta\right)\exp\left(x^\top\hat{\beta}_{\cdot l}\right)}{\exp\left(x_k\delta\right)\sum_{m\leq M}\exp\left(x^\top\hat{\beta}_{\cdot m}\right)} \\
&= \hat{P}_0\left(y = l|x\right)
\end{aligned}$$

Now, as the unpenalized loss is entirely determined by the estimated probabilities and the outcomes, this result tells us that the row means do not affect the unpenalized loss. This is not the case for our penalized problem — here, the row means are all 0.

Lemma 3.1 *For a given X matrix, y vector, and $\lambda > 0$, let β^* denote the solution to the minimization of (4). Let μ^* be the vector of row-means of β^*.*

We have that $\mu^* = \mathbf{0}$.

Proof 3.2 *Let $\beta^{**} = \beta^* - \mathbf{1}\mu^{*\top}$ be the "row-mean centered" version of β^*. Plugging these in to the penalized log-likelihood in (4), we see that the difference between two penalized log-likelihoods is*

$$\begin{aligned}
0 &\leq L\left(\beta^*\right) - L\left(\beta^{**}\right) \\
&= \lambda\sum_{k\leq p}\left[\|\beta^*_{k\cdot}\|_2 - \|\beta^{**}_{k\cdot}\|_2\right] \\
&= \lambda\sum_{k\leq p}\left[\|\beta^*_{k\cdot}\|_2 - \sqrt{\left\|\beta^*_{k\cdot}\right\|_2^2 + \left\|\mu^*_{k\cdot}\right\|_2^2}\right] \\
&\leq 0
\end{aligned}$$

Thus $\|\mu^*_{k\cdot}\|_2 = 0$, so $\mu^* = \mathbf{0}$.

3.2. Multinomial Optimization

This optimization problem is nastier than its Gaussian counterpart, as the coordinate updates no longer have a closed form solution. However, as in `glmnet` we can use an approximate Newton scheme and optimize our multinomial objective by repeatedly approximating with a quadratic and minimizing the corresponding gaussian problem.

We begin with $\tilde{\beta}$, some inital guess of β. From this estimate, we can find an estimate of our probabilities:

$$p_{i,m} = \frac{\exp\left(X_{i\cdot}\tilde{\beta}_{\cdot m}\right)}{\sum_{l\leq M}\exp\left(X_{i\cdot}\tilde{\beta}_{\cdot l}\right)}$$

Now, as in standard Newton-Raphson, we calculate the first and second derivatives of our log-likelihood (in η). We see that

$$\frac{\partial \ell}{\partial \eta_{i,m}} = y_{i,m} - p_{i,m}$$

For the second derivatives, as usual we get "independence" between observations, ie. if $j \neq i$, for any m, l

$$\frac{\partial \ell}{\partial \eta_{i,m}\partial \eta_{j,l}} = 0.$$

We also have our usual Hessian within observation

$$\frac{\partial \ell}{\partial \eta_{i,m}\partial \eta_{i,l}} = p_{i,m}p_{i,l} \tag{5}$$

for $m \neq l$, and

$$\frac{\partial \ell}{\partial \eta_{i,m}^2} = -p_{i,m}(1-p_{i,m}). \tag{6}$$

Let H_i denote the within observation Hessian. By combining (5) and (6) we see that

$$-H_i = \operatorname{diag}(p_{i\cdot}) - p_{i\cdot}^\top p_{i\cdot}$$

and we can write out a second order Taylor series approximation to our log-likelihood (centered around some value $\tilde{\beta}$) by

$$\begin{aligned} -\ell(\beta) \approx -\ell(\tilde{\beta}) - \operatorname{trace}\left[(\beta-\tilde{\beta})^\top X^\top (Y-P)\right] \\ -\frac{1}{2}\sum_{i=1}^{n} X_{i\cdot}\left(\beta-\tilde{\beta}\right) H_i \left(\beta-\tilde{\beta}\right)^\top X_{i\cdot}^\top \end{aligned}$$

Because we have independence of observations, the second order term has decoupled into the sum of simple quadratic forms. Unfortunately, because each of these H_i are different (and not even full rank), using this quadratic approximation instead of the original log-likelihood would still be difficult. We would like to find a simple matrix which dominates all of the $-H_i$. To this end, we show that $-H_i \preceq tI$ where

$$t = 2\max_{i,j}\left\{p_{i,j}\left(1-p_{i,j}\right)\right\} \leq 1/2$$

Lemma 3.3 *Let* $\mathbf{p}$ *be an* M*-vector of probabilities and define*

$$-H = \operatorname{diag}(\mathbf{p}) - \mathbf{p}\mathbf{p}^\top.$$

We have that $-H \preceq \max_l [2\mathbf{p}_l(1-\mathbf{p}_l)]\, I$

Proof 3.4 *Define* D *by*

$$D = \operatorname{diag}[2\mathbf{p}(1-\mathbf{p})] + H$$

We would like to show that D *diagonally dominant and thus positive semi-definite. Toward this end, choose some* $m \leq M$*.*

$$\begin{aligned}\sum_{l\neq m} |D_{m,l}| &= \sum_{l\neq m} \mathbf{p}_l \mathbf{p}_m \\ &= \mathbf{p}_m \sum_{l\neq m} \mathbf{p}_l \\ &= \mathbf{p}_m (1-\mathbf{p}_m) \\ &= D_{m,m}\end{aligned}$$

Thus D *is positive semi-definite, and so* $-H \preceq \operatorname{diag}[2\mathbf{p}(1-\mathbf{p})]$*. Furthermore,* $\operatorname{diag}[2\mathbf{p}(1-\mathbf{p})] \preceq \max_l [2\mathbf{p}_l(1-\mathbf{p}_l)]\, I$ *so we also have*

$$-H \preceq \max_l [2\mathbf{p}_l(1-\mathbf{p}_l)]\, I$$

Now if we consider $t_i = 2\max_j \{p_{i,j}(1-p_{i,j})\}$, then the preceeding lemma gives us that $-H_i \preceq t_i I$, and since for all i $t_i \leq t$, we have our majorization $-H_i \preceq tI$.

If we replace our original Hessian with this majorizing approximation, with some algebraic manipulation we can reduce our problem to the gaussian framework. Furthermore, because our new approximation dominates the Hessian, we still enjoy nice convergence properties. Toward this end, we write a new majorizing quadratic approximation (by replacing each $-H_i$ by tI)

$$\begin{aligned}-\ell(\beta) \leq -\ell(\tilde{\beta}) &- \operatorname{trace}\left[(\beta-\tilde{\beta})^\top X^\top (Y-P)\right] \\ &+ \frac{t}{2}\operatorname{trace}\left[\sum_{i=1}^{n} \left(\beta-\tilde{\beta}\right)^\top X_{i\cdot}^\top X_{i\cdot}\left(\beta-\tilde{\beta}\right)\right]\end{aligned}$$

which, with simple algebra, becomes

$$-\ell(\beta) \leq -\ell(\tilde{\beta}) - \operatorname{trace}\left[(\beta-\tilde{\beta})^\top X^\top (Y-P)\right] + \frac{t}{2}\left\|X\left(\beta-\tilde{\beta}\right)\right\|_F^2$$

Completing the square, we see that minimizing this with the addition of our penalty is equivalent to

$$\min_\beta \frac{1}{2}\left\|X\left(\beta-\tilde{\beta}\right) - (Y-P)/t\right\|_F^2 + \lambda/t \sum \|\beta_{i\cdot}\|_2$$

Notice that we have reduced the multinomial problem to our gaussian framework. Our plan of attack is to repeatedly approximate our loss by this penalized quadratic (centered at our

current estimate of β), and minimize this loss to update our estimate of β.

Multinomial Algorithm
Combining the outer loop steps with our gaussian algorithm we have the following simple algorithm. We can still include all the other bells and whistles from `glmnet` as well.

1. Initialize $\beta = \beta_0$
2. Iterate until convergence:
   (a) Update η by $\eta = X\beta$
   (b) Update P by
   $$p_{i,m} = \frac{\exp(\eta_{i,m})}{\sum_{l=1}^{M} \exp(\eta_{i,l})}$$
   (c) Set $t = 2\max_{i,j}\{p_{i,j}(1-p_{i,j})\}$ and set $R = (Y-P)/t$
   (d) Iterate until convergence: for $k = 1, \ldots, p$
      i. Update R_{-k} by
      $$R_{-k} = R + X_{\cdot k}\beta_{k\cdot}$$
      ii. Update $\beta_{k\cdot}$ by
      $$\beta_{k\cdot} \leftarrow \frac{1}{\|X_{\cdot k}\|_2^2}\left(1 - \frac{\lambda/t}{\left\|X_{\cdot k}^\top R_{-k}\right\|_2}\right)_+ X_{\cdot k}^\top R_{-k}$$
      iii. Update R by
      $$R = R_{-k} - X_{\cdot k}\beta_{k\cdot}$$

3.3. Path Solution

Generally we will be interested in models for more than one value of λ. As per usual in `glmnet` we compute solutions for a path of λ values. We begin with the smallest λ such that $\beta = 0$, and end with λ near 0. By initializing our algorithm for a new λ value at the solution for the previous value, we increase the stability of our algorithm (especially in the presence of correlated features) and efficiently solve along the path. It is straightforward to see that our first λ value is

$$\lambda_{\max} = \max_k \left\|X_{\cdot k}^\top (Y - P_0)\right\|_2$$

where P_0 is just a matrix of the sample proportions in each class. We generally do not solve all the way to the unregularized end of the path. When λ is near 0 the solution is very poorly statistically behaved and the algorithm takes a long time to converge — any reasonable model selection criterion will choose a more restricted model. To that end, we choose $\lambda_{\min} = \epsilon\lambda_{\max}$ (with $\epsilon = 0.05$ in our implementation) and compute solutions over a grid of m values with $\lambda_j = \lambda_{\max}(\lambda_{\min}/\lambda_{\max})^{j/m}$ for $j = 0, \ldots, m$.

3.4. Strong Rules

It has been demonstrated in Tibshirani, Bien, Friedman, Hastie, Simon, Taylor, and Tibshirani (2012) that using a prescreen can significantly cut down on the computation required for

fitting lasso-like problems. Using a similar argument as in Tibshirani *et al.* (2012), at a given $\lambda = \lambda_j$ we can screen out variables for which

$$\left\| X_{\cdot k}^\top R\left(\lambda_{j-1}\right) \right\|_2 \leq \alpha\left(2\lambda_j - \lambda_{j-1}\right)$$

where $R\left(\lambda_{j-1}\right) = Y - X\hat{\beta}\left(\lambda_{j-1}\right)$ for the gaussian case and $R\left(\lambda_{j-1}\right) = Y - \hat{P}\left(\lambda_{j-1}\right)$ for the multinomial case. Now, these rules are unfortunately not "safe" (they could possibly throw out features which should be in the fit, though in practice they essentially never do). Thus, at the end of our algorithm we must check the Karush-Kuhn Tucker optimality conditions for all the variables to certify that we have reached the optimum (and potentially add back in variables in violation). In practice, there are very rarely violations.

4. Elastic Net

We have seen that in some cases performance of the lasso can be improved by the addition of an ℓ_2 penalty. This is known as the elastic-net (Zou and Hastie 2005). Suppose now we wanted to solve the elastic-net problem

$$\min \frac{1}{2}\|Y - X\beta\|_F^2 + \lambda\alpha \sum \|\beta_{k\cdot}\|_2 + \frac{\lambda(1-\alpha)}{2}\|\beta\|_F^2$$

We can again solve one row of β at a time. The row-wise solution satisfies

$$\|X_{\cdot k}\|_2^2 \hat{\beta}_{k\cdot}^\top - X_{\cdot k}^\top R_{-k} + \lambda\alpha S(\hat{\beta}_{k\cdot}) + \lambda(1-\alpha)\beta_{k\cdot}^\top = 0$$

Again, simple algebra gives us that

$$\hat{\beta}_{k\cdot} = \frac{1}{\|X_{\cdot k}\|_2^2 + \lambda(1-\alpha)} \left(1 - \frac{\lambda\alpha}{\left\|X_{\cdot k}^\top R_{-k}\right\|_2}\right)_+ X_{\cdot k}^\top R_{-k} \tag{7}$$

Thus, the algorithm to fit the elastic net for multiresponse regression is exactly as before with step (b) replaced by (7). We can similarly apply this to multinomial regression, and replace step (ii) of multinomial algorithm with our new update.

5. Timings

We timed our algorithm on simulated data, and compared it to a similar algorithm and implementation (`glmnet`) for the usual multinomial lasso regression without grouping. Both of these implementations are written in `R` with the heavy lifting done in `Fortran`. All simulations were run on an Intel Xeon $X5680$, 3.33 ghz processor. Simulations were run with varying numbers of observations n, features p, and classes M for a path of 100 λ-values with ($\lambda_{\min} = 0.05\lambda_{\max}$) averaged over 10 trials. Features were simulated as standard Gaussian random variables with equicorrelation ρ. In all simulations, we set the true β to have iid $N(0, 4/M^2)$ entries in its first 3 rows, and 0 for all other entries.

From Table 1 we can see that our algorithm for the grouped multinomial lasso is a bit slower than the similar algorithm for the usual multinomial lasso. However, it can still solve very large problems quickly, solving gene-expression sized problems in under a minute.

	$\rho = 0$	$\rho = 0.2$
	$n = 50,\ p = 100,\ M = 5$	
grouped	0.32	0.37
ungrouped	0.14	0.12
	$n = 100,\ p = 1000,\ M = 5$	
grouped	1.01	1.19
ungrouped	0.57	0.46
	$n = 100,\ p = 5000,\ M = 10$	
grouped	3.29	4.23
ungrouped	2.48	2.08
	$n = 200,\ p = 10000,\ M = 10$	
grouped	10.27	14.02
ungrouped	8.06	6.60

Table 1: Timings in seconds for grouped and ungrouped multinomial lasso, for a path of 100 λ-values, averaged over 10 trials, for a variety of n, p, M, and ρ.

6. Discussion

We have given an efficient group descent algorithm for fitting the group-penalized multiresponse and multinomial lasso models and empirically shown the efficiency of our algorithm. It has also been included in the current version (1.8-2) of the `R` package `glmnet`, which to our knowledge is the first publically available implementation for fitting these models.

References

Argyriou A, Evgeniou T, Pontil M (2007). "Multi-task feature learning." In *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference*, volume 19, p. 41. The MIT Press.

Friedman JH, Hastie T, Tibshirani R (2010). "Regularization Paths for Generalized Linear Models via Coordinate Descent." *Journal of Statistical Software*, **33**(1), 1–22. ISSN 1548-7660. URL http://www.jstatsoft.org/v33/i01.

Obozinski G, Tasker B, Jordan M (2007). "Joint covariate selection for grouped classifcation." *Technical report*, University of California, Berkeley.

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

Tibshirani R, Bien J, Friedman J, Hastie T, Simon N, Taylor J, Tibshirani R (2012). "Strong rules for discarding predictors in lasso-type problems." *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*.

Vincent M, Hansen N (2012). "Sparse group lasso and high dimensional multinomial classification." *Arxiv preprint arXiv:1205.1245*.

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

Zou H, Hastie T (2005). "Regularization and Variable Selection via the Elastic Net." *Journal of the Royal Statistical Society B*, **67**(2), 301–320.

Affiliation:

Noah Simon
Sequoia Hall
390 Serra Mall
Stanford University
Stanford, CA 94305
E-mail: nsimon@stanford.edu

Journal of Statistical Software http://www.jstatsoft.org/
published by the American Statistical Association http://www.amstat.org/
Volume VV, Issue II *Submitted:* yyyy-mm-dd
MMMMMM YYYY *Accepted:* yyyy-mm-dd