1 Appendix

1.1 K-fold Cross Validation

We can perform joint optimization for K-fold cross validation by reformulating the problem. Let $(\boldsymbol{y}, \boldsymbol{X})$ be the full data set. We denote the kth fold as $(\boldsymbol{y}_k, \boldsymbol{X}_k)$ and its complement as $(\boldsymbol{y}_{-k}, \boldsymbol{X}_{-k})$. Then the objective of this joint optimization problem is the average validation cost across all K folds:

$$\begin{gathered} \arg\min_{\boldsymbol{\lambda}\in\Lambda} \frac{1}{K}\sum_{k=1}^{K} L(\boldsymbol{y}_k, f_{\hat{\boldsymbol{\theta}}^{(k)}(\boldsymbol{\lambda})}(\boldsymbol{X}_k)) \\ \text{s.t. } \hat{\boldsymbol{\theta}}^{(k)}(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{\theta}\in\Theta} L(\boldsymbol{y}_{-k}, f_{\boldsymbol{\theta}}(\boldsymbol{X}_{-k})) + \sum_{i=1}^{J} \lambda_i P_i(\boldsymbol{\theta}) \text{ for } k = 1, ..., K \end{gathered} \tag{1}$$

1.2 Proof of Theorem 1

Proof. We will show that for a given $\boldsymbol{\lambda}_0$ that satisfies the given conditions, the validation loss is continuously differentiable within some neighborhood of $\boldsymbol{\lambda}_0$. It then follows that if the theorem conditions hold true for almost every $\boldsymbol{\lambda}$, then the validation loss is continuously differentiable with respect to $\boldsymbol{\lambda}$ at almost every $\boldsymbol{\lambda}$.

Suppose the theorem conditions are satisfied at $\boldsymbol{\lambda}_0$. Let $\boldsymbol{B}'$ be an orthonormal set of basis vectors that span the differentiable space $\Omega^{L_T}(\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0), \boldsymbol{\lambda}_0)$ with the subset of vectors $\boldsymbol{B}$ that span the model parameter space.

Let $\tilde{L}_T(\boldsymbol{\theta}, \boldsymbol{\lambda})$ be the gradient of $L_T(\cdot, \boldsymbol{\lambda})$ at $\boldsymbol{\theta}$ with respect to the basis $\boldsymbol{B}$:

$$\tilde{L}_T(\boldsymbol{\theta}, \boldsymbol{\lambda}) =_{\boldsymbol{B}} \nabla L_T(\cdot, \boldsymbol{\lambda})|_{\boldsymbol{\theta}} \tag{2}$$

Since $\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0)$ is the minimizer of the training loss, the gradient of $L_T(\cdot, \boldsymbol{\lambda}_0)$ with respect to the basis $\boldsymbol{B}$ must be zero at $\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0)$:

$${}_{\boldsymbol{B}}\nabla L_T(\cdot, \boldsymbol{\lambda}_0)|_{\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0)} = \tilde{L}_T(\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0), \boldsymbol{\lambda}_0) = 0 \tag{3}$$

From our assumptions, we know that there exists a neighborhood W containing $\boldsymbol{\lambda}_0$ such that $\tilde{L}_T$ is continuously differentiable along directions in the differentiable space $\Omega^{L_T}(\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0), \boldsymbol{\lambda}_0)$. Also, the Jacobian matrix $D\tilde{L}_T(\cdot, \boldsymbol{\lambda}_0)|_{\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0)}$ with respect to basis $\boldsymbol{B}$ is nonsingular. Therefore, by the implicit function theorem, there exist open sets $U \subseteq W$ containing $\boldsymbol{\lambda}_0$ and V containing $\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0)$ and a continuously differentiable function $\gamma : U \to V$ such that for every $\boldsymbol{\lambda} \in U$, we have that

$$\tilde{L}_T(\gamma(\boldsymbol{\lambda}), \boldsymbol{\lambda}) = \nabla_B L_T(\cdot, \boldsymbol{\lambda})|_{\gamma(\boldsymbol{\lambda})} = 0 \tag{4}$$

That is, we know that $\gamma(\boldsymbol{\lambda})$ is a continuously differentiable function that minimizes $L_T(\cdot, \boldsymbol{\lambda})$ in the differentiable space $\Omega^{L_T}(\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0), \boldsymbol{\lambda}_0)$. Since we assumed that the differentiable space is a local optimality space of $L_T(\cdot, \boldsymbol{\lambda})$ in the neighborhood W, then for every $\boldsymbol{\lambda} \in U$,

$$\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}) = \arg\min_{\boldsymbol{\theta}} L_T(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \arg\min_{\boldsymbol{\theta}\in\Omega^{L_T}(\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0), \boldsymbol{\lambda}_0)} L_T(\boldsymbol{\theta}, \boldsymbol{\lambda}) = \gamma(\boldsymbol{\lambda}) \tag{5}$$

Therefore, we have shown that if $\boldsymbol{\lambda}_0$ satisfies the assumptions given in the theorem, the fitted model parameters $\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda})$ is a continuously differentiable function within a neighborhood of $\boldsymbol{\lambda}_0$. We can then apply the chain rule to get the gradient of the validation loss. □

1.3 Regression Examples

1.3.1 Elastic Net

We show that the joint optimization problem for the Elastic Net satisfies all three conditions in Theorem 1:

> Condition 1: The elastic net solution paths are piecewise linear (Zou & Hastie 2003, Tibshirani et al. 2013), which means that the nonzero indices of the elastic net estimates stay locally constant for almost every $\boldsymbol{\lambda}$. Therefore, $S_{\boldsymbol{\lambda}}$ is a local optimality space for $L_T(\cdot, \boldsymbol{\lambda})$. ✓
>
> Condition 2: The ℓ_1 penalty is smooth when restricted to $S_{\boldsymbol{\lambda}}$. ✓
>
> Condition 3: The Hessian matrix of $L_T(\cdot, \boldsymbol{\lambda})$ with respect to the columns of $\boldsymbol{I}_{I(\boldsymbol{\lambda})}$ is $\boldsymbol{I}_{I(\boldsymbol{\lambda})}^\top \boldsymbol{X}_T^\top \boldsymbol{X}_T \boldsymbol{I}_{I(\boldsymbol{\lambda})} + \lambda_2 \boldsymbol{I}$. This is positive definite if $\lambda_2 > 0$. ✓

1.3.2 Additive Models with Sparsity and Smoothness Penalties

Let

$$\boldsymbol{U} = \begin{bmatrix} \boldsymbol{U}^{(i_1)} & \ldots & \boldsymbol{U}^{(i_{|J(\boldsymbol{\lambda})|})} \end{bmatrix} \tag{6}$$

where $i_\ell \in J(\boldsymbol{\lambda})$. Then

$$\boldsymbol{H}(\boldsymbol{\lambda}) = \boldsymbol{U}^\top \boldsymbol{I}_T^\top \boldsymbol{I}_T \boldsymbol{U} + \lambda_0 \, diag\left(\frac{1}{||\boldsymbol{U}^{(i)} \hat{\boldsymbol{\beta}}^{(i)}(\boldsymbol{\lambda})||_2} \left(\boldsymbol{I} - \frac{\hat{\boldsymbol{\beta}}^{(i)}(\boldsymbol{\lambda})^\top \hat{\boldsymbol{\beta}}^{(i)}(\boldsymbol{\lambda})}{||\boldsymbol{U}^{(i)} \hat{\boldsymbol{\beta}}^{(i)}(\boldsymbol{\lambda})||_2} \right) \right) + \epsilon \boldsymbol{I} \tag{7}$$

Now we check that all three conditions are satisfied.

> Condition 1: It seems likely that the space spanned by the nonzero coefficients of $\boldsymbol{\theta}$ is a local optimality space, though we are unable to formally prove this fact. Empirically, it has been found that the group lasso solution paths are smooth almost everywhere (Yuan & Lin 2006). On the theoretical side, Vaiter et al. (2012) proved that the active set in a group lasso problem is locally constant for small perturbations in the response. Similar techniques can probably be used to show that the active set is locally constant for small perturbations in the penalty parameters. ✓?
>
> Condition 2: The ℓ_1 and ℓ_2 penalties are smooth when restricted to $S_{\boldsymbol{\lambda}}$. ✓
>
> Condition 3: The Hessian matrix in (7) is positive definite for any $\epsilon > 0$. ✓

The matrix $C(\boldsymbol{\beta}(\boldsymbol{\lambda}))$ in (29) is defined as

$$C(\boldsymbol{\beta}(\boldsymbol{\lambda})) = \begin{cases} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U}^{(i)\top} \boldsymbol{D}_{\boldsymbol{x}_i}^{(2)\top} sgn(\boldsymbol{D}_{\boldsymbol{x}_i}^{(2)} \boldsymbol{U}^{(i)} \hat{\boldsymbol{\beta}}^{(i)}) \\ \boldsymbol{0} \end{bmatrix} & \text{for } i \in J(\boldsymbol{\lambda}) \\ \boldsymbol{0} & \text{for } i \notin J(\boldsymbol{\lambda}) \end{cases} \tag{8}$$

1.3.3 Un-pooled Sparse Group Lasso

The Hessian in this problem is

$$\boldsymbol{H}(\boldsymbol{\lambda}) = \frac{1}{n}\boldsymbol{X}_{T,I(\boldsymbol{\lambda})}^{\top}\boldsymbol{X}_{T,I(\boldsymbol{\lambda})} + diag\left(\frac{\lambda_m}{||\boldsymbol{\theta}^{(m)}||_2}\left(\boldsymbol{I} - \frac{\boldsymbol{\theta}^{(m)}\boldsymbol{\theta}^{(m)\top}}{||\boldsymbol{\theta}^{(m)}||_2^2}\right)\right) + \epsilon\boldsymbol{I} \tag{9}$$

The logic for checking all three conditions in Theorem 1 is similar to the other examples:

Condition 1: The space spanned by the nonzero coefficients of $\boldsymbol{\theta}$ is clearly also a differentiable space. We hypothesize that this space is also a local optimality space $\boldsymbol{\lambda}$, though we have not formally proven this fact. We suspect this to be true for the same reasons discussed in Section 1.3.2.

Condition 2: The ℓ_1 and ℓ_2 penalties are twice-differentiable when restricted to $S_{\boldsymbol{\lambda}}$. ✓

Condition 3: The Hessian matrix in (9) is positive definite for any $\epsilon > 0$. ✓

The matrix $\boldsymbol{C}(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))$ in (33) has columns $m = 1, 2..., M$

$$\begin{bmatrix} \mathbf{0} \\ \frac{\hat{\boldsymbol{\beta}}^{(m)}(\boldsymbol{\lambda})}{||\hat{\boldsymbol{\beta}}^{(m)}(\boldsymbol{\lambda})||_2} \\ \mathbf{0} \end{bmatrix} \tag{10}$$

where $\mathbf{0}$ are the appropriate dimensions.

1.3.4 Low-rank Matrix Completion

We first show that a valid differentiable space of the training criterion (35) is where the rank of the interaction matrix Γ stays constant. Suppose the fitted interaction matrix Γ has SVD decomposition $U\Sigma V^{\top}$ where the i-th singular value is denoted σ_i. The subdifferential of the nuclear norm is

$$\partial\|\Gamma\|_* = \left\{ Udiag(\mu)V^{\top} \middle| \mu_i \in \begin{cases} [-1,1] & \sigma_i = 0 \\ sign(\sigma_i) & \sigma_i \neq 0 \end{cases} \right\} \tag{11}$$

The subdifferential reduces to a gradient if we differentiate with respect to matrices of rank no larger than the rank of Γ.

To derive the gradient of the validation error with respect to the penalty parameters, we perform implicit differentiation of the gradient optimality conditions to get a system of linear equations. However one must be careful in determining the gradient conditions. In particular, we need to transform the subgradient optimality conditions to get gradient optimality conditions. Let the SVD decomposition of $\hat{\boldsymbol{\Xi}}(\boldsymbol{\lambda})$ be $\hat{\boldsymbol{U}}(\boldsymbol{\lambda})\hat{\boldsymbol{\Sigma}}(\boldsymbol{\lambda})\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^{\top}$. After taking the subgradient of the training criterion with respect to $\boldsymbol{\Xi}$, we must multiply the result by its left singular vectors to get the following gradient optimality condition:

$$\mathbf{0} = -\frac{1}{|T|}\hat{\boldsymbol{U}}(\boldsymbol{\lambda})^{\top}\left(\boldsymbol{M} - \boldsymbol{X}_{I_r(\boldsymbol{\lambda})}\hat{\boldsymbol{\eta}}(\boldsymbol{\lambda})\mathbf{1}^{\top} - (\boldsymbol{Z}_{I_c(\boldsymbol{\lambda})}\hat{\boldsymbol{\gamma}}(\boldsymbol{\lambda})\mathbf{1}^{\top})^{\top} - \hat{\boldsymbol{\Xi}}\right) \tag{12}$$

$$+ \lambda_0 sign(\hat{\boldsymbol{\Sigma}}(\boldsymbol{\lambda}))\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^{\top} + \epsilon\hat{\boldsymbol{\Sigma}}(\boldsymbol{\lambda})\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^{\top} \tag{13}$$

Similarly, we multiply the result by its right singular vectors to get

$$\mathbf{0} = -\frac{1}{|T|}\left(\boldsymbol{M} - \boldsymbol{X}_{I_r(\boldsymbol{\lambda})}\hat{\boldsymbol{\eta}}(\boldsymbol{\lambda})\mathbf{1}^\top - (\boldsymbol{Z}_{I_c(\boldsymbol{\lambda})}\hat{\boldsymbol{\gamma}}(\boldsymbol{\lambda})\mathbf{1}^\top)^\top - \hat{\boldsymbol{\Xi}}\right)\hat{\boldsymbol{V}}(\boldsymbol{\lambda}) \tag{14}$$

$$+ \lambda_0\hat{\boldsymbol{U}}(\boldsymbol{\lambda})sign(\hat{\boldsymbol{\Sigma}}(\boldsymbol{\lambda})) + \epsilon\hat{\boldsymbol{\Sigma}}(\boldsymbol{\lambda})\hat{\boldsymbol{V}}(\boldsymbol{\lambda}) \tag{15}$$

To get additional linear constraints, we implicitly differentiate the conditions $\hat{\boldsymbol{U}}^\top(\boldsymbol{\lambda})\hat{\boldsymbol{U}}(\boldsymbol{\lambda}) = \boldsymbol{I}$ and $\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^\top\hat{\boldsymbol{V}}(\boldsymbol{\lambda}) = \boldsymbol{I}$ with respect to $\boldsymbol{\lambda}$. Combining this with the gradient optimality conditions with respect to η and γ, we can derive the gradient of the validation loss with respect to $\boldsymbol{\lambda}$. Note that the solution to this system of linear equations will give us the partial derivatives $\frac{\partial}{\partial\boldsymbol{\lambda}}\hat{\boldsymbol{U}}(\boldsymbol{\lambda})$, $\frac{\partial}{\partial\boldsymbol{\lambda}}\hat{\boldsymbol{\Sigma}}(\boldsymbol{\lambda})$, and $\frac{\partial}{\partial\boldsymbol{\lambda}}\hat{\boldsymbol{V}}(\boldsymbol{\lambda})$. From this, we can derive $\frac{\partial}{\partial\boldsymbol{\lambda}}\hat{\boldsymbol{\Xi}}(\boldsymbol{\lambda})$.

Finally, we check the three conditions in Theorem 1. The reasoning is is similar to the other examples:

Condition 1: In Section 2.4.4, we showed that the differentiable space of the joint optimization problem at $\boldsymbol{\lambda}$ is the space of matrices are the same rank as $\hat{\boldsymbol{\Gamma}}(\hat{\boldsymbol{\lambda}})$ and the space spanned by the nonzero row and column groups of $\hat{\boldsymbol{\alpha}}(\boldsymbol{\lambda})$ and $\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda})$. We hypothesize that this is also a local optimality space $\boldsymbol{\lambda}$. This seems to be justified for the same reasons mentioned in 1.3.2. In addition, it has been shown empirically that for matrix completion problems with a nuclear norm penalty, similar penalty parameter result in fitted matrices of similar rank (Mazumder et al. 2010).

Condition 2: The nuclear norm penalty and the group lasso penalties are twice-differentiable when restricted to $S_{\boldsymbol{\lambda}}$. ✓

Condition 3: The Hessian of the training criterion is positive definite for any $\epsilon > 0$ since the nuclear norm and the group lasso penalties are convex and the frobenius norm and the ridge penalties are positive definite. ✓

1.4 Backtracking Line Search

Let the criterion function be $L : \mathbb{R}^n \to \mathbb{R}$. Suppose that the descent algorithm is currently at point x with descent direction Δx. Backtracking line search uses a heuristic for finding a step size $t \in (0, 1]$ such that the value of the criterion is minimized. The method depends on constants $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$.

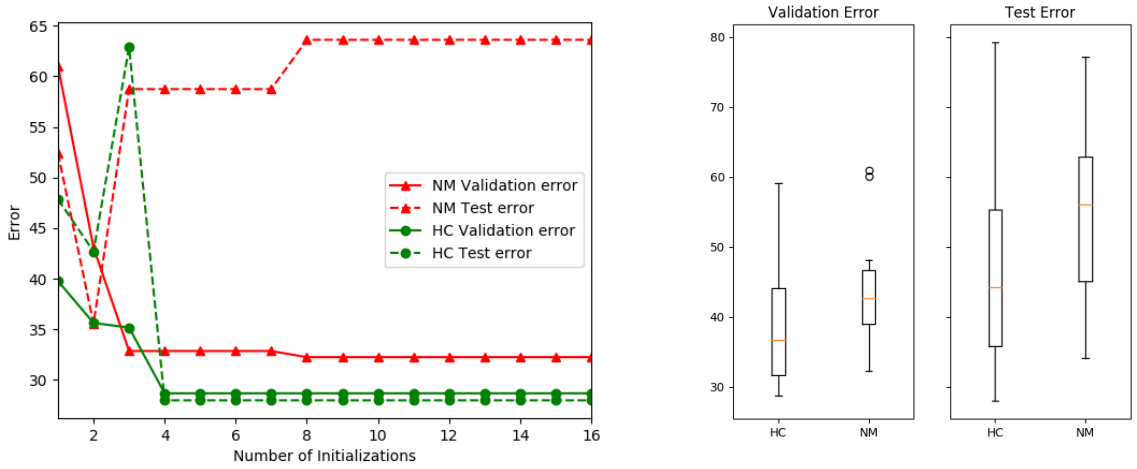
Algorithm Backtracking Line Search

Initialize $t = 1$.
while $L(\boldsymbol{x} + t\boldsymbol{\Delta x}) > L(\boldsymbol{x}) + \alpha t\nabla L(\boldsymbol{x})^T\boldsymbol{\Delta x}$ **do**
  Update $t := \beta t$
end while

Table 1: Additive models fitted with two penalty parameters, tuned by gradient descent, Nelder-Mead, and Spearmint. Standard errors are given in parentheses.

	Num λ	Validation Error	Test Error	# Solves
Gradient Descent	2	23.87 (0.97)	26.10 (0.86)	13.07
Nelder-Mead	2	28.86 (1.04)	29.97 (0.96)	100
Spearmint	2	29.18 (1.07)	30.09 (1.08)	100

Figure 1: Error of additive models tuned by gradient descent vs. Nelder-mead. Left: Validation and test error of models after as the number of initialization points increases. Right: The distribution of validation and test errors.



1.5 Simulation studies additional tables

1.5.1 Sparse add models

To assess how sensitive gradient descent is to its initialization points, we tried multiple starting points for a smaller problem with 60 training, 30 validation, and 30 test observations and $p = 15$ covariates. The response was generated from (39), so the first three functions are non-zero and the remaining 12 are zero. We initialized $\boldsymbol{\lambda}$ by considering all possible combinations of $(\lambda_0, \lambda_1 \mathbf{1})$ where $\lambda_0, \lambda_1 \in \{10^i : i \in \{-2, ..., 1\}\}$. The initializations were randomly shuffled. The results are shown in Figure 1.5.1. For gradient descent, the validation and test error decreased with increasing number of observations, but there are no changes after the fourth initialization point. For Nelder-mead, the validation and test error continue to change as initializations increase. Even though the validation error decreases, the test error actually increases. This is probably due to the fact that Nelder-mead is unable to search the penalty parameter space effectively high-dimensions. Hence it essentially selects penalty parameters at random. From Figure 1.5.1 (Right), we also see that gradient descent fits models with lower validation and test error on average compared to Nelder-Mead.

Table 2: Sparse Group Lasso fitted with two penalty parameters. Standard errors are given in parentheses. We abbreviated the methods as follows: Gradient Descent = GD, Nelder-Mead = NM, Spearmint = SP

n=90, p=600, M=30					
	# λ	β Error	Validation Err	Test Err	# Solves
GD	2	7.37 (0.18)	46.82 (2.21)	49.33 (1.36)	21.43
NM	2	7.31 (0.18)	46.37 (2.24)	48.95 (1.35)	100
SP	2	7.35 (0.20)	45.70 (2.32)	49.35 (1.56)	100
n=90, p=900, M=60					
	# λ	β Error	Validation Error	Test Error	# Solves
GD	2	7.58 (0.21)	45.71 (2.26)	50.31 (1.93)	20.77
NM	2	7.56 (0.19)	44.95 (2.24)	50.18 (1.82)	100
SP	2	8.20 (0.20)	49.59 (2.27)	56.54 (2.14)	100
n=90, p=1200, M=100					
	# λ	β Error	Validation Error	Test Error	# Solves
GD	2	8.27 (0.19)	50.46 (2.30)	57.02 (1.94)	19.80
NM	2	8.09 (0.19)	49.92 (2.33)	55.46 (1.89)	100
SP	2	8.20 (0.20)	49.70 (2.26)	56.51 (2.16)	100

1.5.2 Un-pooled Sparse group lasso

Table 2 displays results from fitting the two-parameter version of the joint optimization problem (30) using gradient descent, Nelder-Mead, and Spearmint. Comparing th results in , we see that all four methods give similar the validation and test errors when tuning the model with two penalty parameters. Therefore regardless of the method used to tune the twp-parameter sparse group lasso, the un-pooled sparse group lasso gives models with significantly lower test error.

1.5.3 Low-rank Matrix Completion

The results from fitting the two-parameter version of the joint optimization problem (35) using gradient descent, Nelder-Mead, and Spearmint are displayed in Table 3. Comparing these results to those in Table , we see that these methods produce similar validation and test errors as grid search. More importantly, these results show that having separate penalty parameters for each of the covariate groups results in lower test error, regardless of the method used to tune the two-parameter joint optimization problem.

References

Mazumder, R., Hastie, T. & Tibshirani, R. (2010), 'Spectral regularization algorithms for learning large incomplete matrices', *Journal of machine learning research* **11**(Aug), 2287–2322.

Table 3: Matrix Completion fitted with two penalty parameters. Standard errors are given in parentheses. We abbreviated the methods as follows: Gradient Descent = GD, Nelder-Mead = NM, Spearmint = SP

	# λ	Validation Err	Test Err	Num Solves
GD	2	0.70 (0.04)	0.71 (0.04)	8.03 (0.79)
NM	2	0.71 (0.04)	0.71 (0.04)	100
SP	2	0.73 (0.04)	0.74 (0.04)	100

Tibshirani, R. J. et al. (2013), 'The lasso problem and uniqueness', *Electronic Journal of Statistics* **7**, 1456–1490.

Vaiter, S., Deledalle, C., Peyré, G., Fadili, J. & Dossal, C. (2012), 'The degrees of freedom of the group lasso', *arXiv preprint arXiv:1205.1481* .

Yuan, M. & Lin, Y. (2006), 'Model selection and estimation in regression with grouped variables', *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **68**(1), 49–67.

Zou, H. & Hastie, T. (2003), 'Regression shrinkage and selection via the elastic net, with applications to microarrays', *Journal of the Royal Statistical Society: Series B. v67* pp. 301–320.