A Appendix

A.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:

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

A.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 $\boldsymbol{B}$:

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

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 $\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}_0)$:

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

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

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

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 $\boldsymbol{\lambda} \in U$,

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

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. □

A.3 Regression Examples

A.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), which means that the nonzero indices of the elastic net estimates stay locally constant for almost every $\boldsymbol{\lambda}$. Therefore, $S_{\boldsymbol{\lambda}}$ as defined in Section 2.4.1 is a local optimality space for $L_T(\cdot, \boldsymbol{\lambda})$. ✓
>
> Condition 2: We only need to establish that the ℓ_1 penalty is twice-continuously differentiable in the directions of $S_{\boldsymbol{\lambda}}$ since the quadratic loss function and the ridge penalty are both smooth. The absolute value function is twice-continuously differentiable everywhere except at zero. Hence the training criterion is smooth when restricted to $S_{\boldsymbol{\lambda}}$. ✓
>
> Condition 3: The Hessian matrix of $L_T(\cdot, \boldsymbol{\lambda})$ with respect to $\boldsymbol{I}_{I(\boldsymbol{\lambda})}$ is $\boldsymbol{I}_{I(\boldsymbol{\lambda})}^\top \boldsymbol{X}_T^\top \boldsymbol{X}_T \boldsymbol{I}_{I(\boldsymbol{\lambda})} + \lambda_2 \boldsymbol{I}$. The first summand is positive semi-definite. As long as $\lambda_2 > 0$, the contribution of the identity matrix ensures the Hessian is positive definite. ✓

A.3.2 Additive Models with Sparsity and Smoothness Penalties

Let

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

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

The gradient of the validation loss is

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{y_V}, f_{\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda})}(\boldsymbol{X_V})) = -\left(\boldsymbol{I}_V \sum_{i \in J(\boldsymbol{\lambda})} \boldsymbol{U}^{(i)} \frac{\partial}{\partial \boldsymbol{\lambda}} \hat{\boldsymbol{\beta}}^{(i)}(\boldsymbol{\lambda}) \right)^\top \left(\boldsymbol{y}_V - \boldsymbol{I}_V \sum_{i \in J(\boldsymbol{\lambda})} \boldsymbol{U}^{(i)} \hat{\boldsymbol{\beta}}^{(i)}(\boldsymbol{\lambda}) \right) \tag{A.7}$$

where

$$\frac{\partial}{\partial \boldsymbol{\lambda}} \begin{bmatrix} \hat{\boldsymbol{\beta}}^{(i_1)}(\boldsymbol{\lambda}) \\ \ldots \\ \hat{\boldsymbol{\beta}}^{(i_{|J(\boldsymbol{\lambda})|})}(\boldsymbol{\lambda}) \end{bmatrix} = \boldsymbol{H}(\boldsymbol{\lambda})^{-1} \begin{bmatrix} \begin{bmatrix} \frac{\hat{\boldsymbol{\beta}}^{(i_1)}(\boldsymbol{\lambda})}{\left\| \boldsymbol{U}^{(i_1)} \hat{\boldsymbol{\beta}}^{(i_1)}(\boldsymbol{\lambda}) \right\|_2} \\ \ldots \\ \frac{\hat{\boldsymbol{\beta}}^{\left(i_{|J(\boldsymbol{\lambda})|}\right)}(\boldsymbol{\lambda})}{\left\| \boldsymbol{U}^{\left(i_{|J(\boldsymbol{\lambda})|}\right)} \hat{\boldsymbol{\beta}}^{\left(i_{|J(\boldsymbol{\lambda})|}\right)}(\boldsymbol{\lambda}) \right\|_2} \end{bmatrix} & \boldsymbol{C}\left(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) \right) \end{bmatrix} \tag{A.8}$$

The Hessian $\boldsymbol{H}(\boldsymbol{\lambda})$ is

$$\boldsymbol{H}(\boldsymbol{\lambda}) = \boldsymbol{U}^\top \boldsymbol{I}_T^\top \boldsymbol{I}_T \boldsymbol{U} + \lambda_0 diag \left(\left\{ \frac{1}{||\boldsymbol{U}^{(i_\ell)} \hat{\boldsymbol{\beta}}^{(i_\ell)}(\boldsymbol{\lambda})||_2} \left(\boldsymbol{I} - \frac{\hat{\boldsymbol{\beta}}^{(i_\ell)}(\boldsymbol{\lambda}) \hat{\boldsymbol{\beta}}^{(i_\ell)\top}(\boldsymbol{\lambda})}{||\boldsymbol{U}^{(i_\ell)} \hat{\boldsymbol{\beta}}^{(i_\ell)}(\boldsymbol{\lambda})||_2^2} \right) \right\}_{i_\ell \in J(\boldsymbol{\lambda})} \right) + \epsilon \boldsymbol{I} \tag{A.9}$$

and $\boldsymbol{C}(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))$ has columns $i = 1, ..., p$

$$\boldsymbol{C}_i(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda})) = \begin{cases} \begin{bmatrix} \boldsymbol{0} \\ \boldsymbol{U}^{(i)\top}\boldsymbol{D}_{\boldsymbol{x}_i}^{(2)\top} sgn\left(\boldsymbol{D}_{\boldsymbol{x}_i}^{(2)}\boldsymbol{U}^{(i)}\hat{\boldsymbol{\beta}}^{(i)}(\boldsymbol{\lambda})\right) \\ \boldsymbol{0} \end{bmatrix} & \text{for } i \in J(\boldsymbol{\lambda}) \\ \boldsymbol{0} & \text{for } i \notin J(\boldsymbol{\lambda}) \end{cases} \tag{A.10}$$

Now we check that all three conditions are satisfied.

Condition 1: It seems likely that the space spanned by $S_{\boldsymbol{\lambda}}$ is a local optimality space, though we are unable to formally prove this. The training criterion for this problem is composed of generalized lasso penalties and a group lasso penalties. For the generalized lasso, Tibshirani et al. (2011) proved that the solution path is smooth almost everywhere. For the group lasso, there is empirical evidence that the active set is locally constant almost everywhere with respect to the penalty parameter (Yuan & Lin 2006), but this has not been formally proven. Vaiter et al. (2012) showed that the active set is locally constant with respect to the response; we suspect similar techniques could be used to prove our hypothesis.

Condition 2: We only need to establish that the generalized lasso and group lasso penalties are twice-continuously differentiable in the directions of $S_{\boldsymbol{\lambda}}$ since the rest of the training criterion is smooth. $\|\boldsymbol{D\theta}\|_1$ is not differentiable at the points where $\boldsymbol{D\theta}$ has zero elements. We must therefore restrict the derivatives to be taken in directions such that the zero elements of $\boldsymbol{D\theta}$ remain constant. The ℓ_2 norm is twice-continuously differentiable everywhere except at the zero vector. Hence the training criterion is smooth when restricted to the differentiable space $S_{\boldsymbol{\lambda}}$ specified in Section 2.4.2. ✓

Condition 3: The Hessian matrix in (A.9) is the sum of positive semi-definite matrices. As long as $\epsilon > 0$, the contribution of the last summand $\epsilon \boldsymbol{I}$ will make the Hessian matrix positive-definite. ✓

A.3.3 Un-pooled Sparse Group Lasso

The gradient of the validation loss with respect to the penalty parameters is

$$\nabla_{\boldsymbol{\lambda}} L(\boldsymbol{y_V}, f_{\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda})}(\boldsymbol{X_V})) = -\left(\boldsymbol{X}_{V,I(\boldsymbol{\lambda})}\frac{\partial}{\partial \boldsymbol{\lambda}}\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda})\right)^\top \left(\boldsymbol{y}_V - \boldsymbol{X}_{V,I(\boldsymbol{\lambda})}\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda})\right) \tag{A.11}$$

where

$$\frac{\partial}{\partial \boldsymbol{\lambda}}\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}) = -\boldsymbol{H}(\boldsymbol{\lambda})^{-1}\left[\boldsymbol{C}(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda})) \quad sgn(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))\right] \tag{A.12}$$

The Hessian $\boldsymbol{H}(\boldsymbol{\lambda})$ is

$$\boldsymbol{H}(\boldsymbol{\lambda}) = \frac{1}{n}\boldsymbol{X}_{T,I(\boldsymbol{\lambda})}^\top \boldsymbol{X}_{T,I(\boldsymbol{\lambda})} + diag\left(\frac{\lambda_m}{||\boldsymbol{\theta}^{(m)}||_2}\left(\boldsymbol{I} - \frac{\boldsymbol{\theta}^{(m)}\boldsymbol{\theta}^{(m)\top}}{||\boldsymbol{\theta}^{(m)}||_2^2}\right)\right) + \epsilon \boldsymbol{I} \tag{A.13}$$

The matrix $\boldsymbol{C}(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda}))$ in (A.12) has columns $m = 1, 2..., M$

$$\boldsymbol{C}_i(\hat{\boldsymbol{\beta}}(\boldsymbol{\lambda})) = \begin{bmatrix} \boldsymbol{0} \\ \frac{\hat{\boldsymbol{\beta}}^{(m)}(\boldsymbol{\lambda})}{||\hat{\boldsymbol{\beta}}^{(m)}(\boldsymbol{\lambda})||_2} \\ \boldsymbol{0} \end{bmatrix} \tag{A.14}$$

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

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

Condition 1: We hypothesize that the differentiable space $S_{\boldsymbol{\lambda}}$ is also a local optimality space, though we have not formally proven this fact. We suspect this is true for the same reasons discussed in Section A.3.2.

Condition 2: The ℓ_1 and ℓ_2 penalties are twice-differentiable when restricted to $S_{\boldsymbol{\lambda}}$ for the same reasons discussed in Section A.3.2. ✓

Condition 3: The Hessian matrix in (A.13) is the sum of positive semi-definite matrices. It is positive definite for any $\epsilon > 0$ due to the last summand $\epsilon \boldsymbol{I}$. ✓

A.3.4 Low-rank Matrix Completion

Here we derive the differentiable space of the training criterion with respect to $\boldsymbol{\Gamma}$. At $\boldsymbol{\lambda}$, suppose the fitted interaction matrix $\hat{\boldsymbol{\Gamma}}(\boldsymbol{\lambda})$ has a singular value decomposition $\hat{\boldsymbol{U}}(\boldsymbol{\lambda})\text{diag}(\hat{\boldsymbol{\sigma}}(\boldsymbol{\lambda}))\hat{\boldsymbol{V}}^\top(\boldsymbol{\lambda})$. We denote the ith singular value/vector with subscript i. Then the differentiable space with respect to $\boldsymbol{\Gamma}$ at $\hat{\boldsymbol{\Gamma}}(\boldsymbol{\lambda})$ is

$$S_{\boldsymbol{\lambda},\boldsymbol{\Gamma}} = \left\{ \boldsymbol{B} \in \mathbb{R}^{N \times N} \middle| \hat{\boldsymbol{U}}_i^\top(\boldsymbol{\lambda}) \boldsymbol{B} \hat{\boldsymbol{V}}_i(\boldsymbol{\lambda}) = 0 \quad \forall i \text{ s.t. } \sigma_i = 0 \right\} \tag{A.15}$$

$$= \text{span}\left(\left\{ \hat{\boldsymbol{U}}_i(\boldsymbol{\lambda}) \boldsymbol{b}_u^\top + \boldsymbol{b}_v \hat{\boldsymbol{V}}_i^\top(\boldsymbol{\lambda}) \middle| \boldsymbol{b}_u, \boldsymbol{b}_v \in \mathbb{R}^N, \sigma_i \neq 0 \right\} \right) \tag{A.16}$$

The proof is a direct application of Theorem 1 in Watson (1992). The following lemma adapts his results for our purposes. Note that if a matrix can be written as a univariate function $\tilde{\boldsymbol{\Gamma}}(\epsilon)$, its singular values and singular vectors can be numbered such that they are each a function of ϵ, e.g. $\sigma_i(\epsilon)$, $\boldsymbol{U}_i(\epsilon)$, and $\boldsymbol{V}_i(\epsilon)$ (Rellich 1969).

Lemma 1. *Suppose $\boldsymbol{\Gamma} \in \mathbb{R}^{N \times N}$ has a singular value decomposition $\boldsymbol{U}\,\text{diag}(\boldsymbol{\sigma})\boldsymbol{V}$. Let*

$$\mathcal{B} = \left\{ \boldsymbol{B} \in \mathbb{R}^{N \times N} \middle| \boldsymbol{U}_i^\top \boldsymbol{B} \boldsymbol{V}_i = 0 \quad \forall i \ s.t.\ \sigma_i = 0 \right\} \tag{A.17}$$

The directional derivative of the nuclear norm $\|\cdot\|_*$ at $\boldsymbol{\Gamma}$ along $\boldsymbol{B} \in \mathcal{B}$ is

$$\lim_{\epsilon \to 0^+} \frac{\|\boldsymbol{\Gamma} + \epsilon \boldsymbol{B}\|_* - \|\boldsymbol{\Gamma}\|_*}{\epsilon} = \sum_{i=1}^N \boldsymbol{U}_i^\top \boldsymbol{B} \boldsymbol{V}_i 1_{[\sigma_i \neq 0]} \tag{A.18}$$

Moreover, let the eigenvalues be numbered such that $\sigma_{i,\boldsymbol{B}}(\epsilon)$ denotes the ith singular value of $\boldsymbol{\Gamma} + \epsilon \boldsymbol{B}$. Then

$$\mathcal{B} = \left\{ \boldsymbol{B} \in \mathbb{R}^{N \times N} \middle| \left. \frac{d\sigma_{i,\boldsymbol{B}}(\epsilon)}{d\epsilon} \right|_{\epsilon=0} = 0 \quad \forall i \ s.t.\ \sigma_i = 0 \right\} \tag{A.19}$$

Now we derive the gradient of the validation loss with respect to the penalty parameters. One approach would be to follow Algorithm 2 exactly, which requires us to find an orthonormal basis of (A.16). An alternative approach is to use the result in (A.19): the differentiable space is the set of directions where the zero singular values remain locally constant. Assuming Condition 1 holds, we only need to consider interaction matrices with rank at most $r = \text{rank}(\hat{\boldsymbol{\Gamma}}(\boldsymbol{\lambda}))$. Hence a locally equivalent training criterion is:

$$\begin{aligned} \underset{\substack{\boldsymbol{\eta},\boldsymbol{\gamma} \\ \boldsymbol{\Gamma}=\boldsymbol{U}diag(\boldsymbol{\sigma})\boldsymbol{V}^\top \\ \boldsymbol{U},\boldsymbol{V}\in\mathbb{R}^{N\times r},\boldsymbol{\sigma}\in\mathbb{R}^r}}{\arg\min} \quad & \frac{1}{2}\left\|\boldsymbol{M} - \boldsymbol{X}_{I_r(\boldsymbol{\lambda})}\boldsymbol{\eta}\boldsymbol{1}^\top - (\boldsymbol{Z}_{I_c(\boldsymbol{\lambda})}\boldsymbol{\gamma}\boldsymbol{1}^\top)^\top - \boldsymbol{\Gamma}\right\|_T^2 + \lambda_0\left\|\boldsymbol{\Gamma}\right\|_* \\ & + \sum_{g=1}^G \lambda_g\|\boldsymbol{\eta}^{(g)}\|_2 + \sum_{g=1}^G \lambda_{G+g}\|\boldsymbol{\gamma}^{(g)}\|_2 + \frac{1}{2}\epsilon\left(\|\boldsymbol{\eta}\|_2^2 + \|\boldsymbol{\gamma}\|_2^2 + \|\boldsymbol{\Gamma}\|_F^2\right) \end{aligned} \tag{A.20}$$

$$\text{s.t. } \boldsymbol{V}^\top\boldsymbol{V} = \boldsymbol{I} \text{ and } \boldsymbol{U}^\top\boldsymbol{U} = \boldsymbol{I} \tag{A.21}$$

The locally equivalent training criterion is now smooth at its minimizer. The gradient optimality conditions with respect to $\boldsymbol{\Gamma}$ can be taken with respect to the basis

$$\left\{\hat{\boldsymbol{U}}_i(\boldsymbol{\lambda})\boldsymbol{e}_j^\top | i = 1, ..., r; j = 1, ..., N\right\} \cup \left\{\boldsymbol{e}_j\hat{\boldsymbol{V}}_i(\boldsymbol{\lambda})^\top | i = 1, ..., r; j = 1, ..., N\right\} \tag{A.22}$$

Note that this basis is quite different from that used in Algorithm 2; it is allowed to vary with $\boldsymbol{\lambda}$ and its elements are not orthonormal. The benefit of this alternative approach is that the gradient optimality condition for $\boldsymbol{\Gamma}$ is easy to derive. Taking the gradient with respect to the directions in A.22, we get:

$$\begin{aligned} \boldsymbol{0} = & -\hat{\boldsymbol{U}}(\boldsymbol{\lambda})^\top\left(\boldsymbol{M} - \boldsymbol{X}_{I_r(\boldsymbol{\lambda})}\hat{\boldsymbol{\eta}}(\boldsymbol{\lambda})\boldsymbol{1}^\top - (\boldsymbol{Z}_{I_c(\boldsymbol{\lambda})}\hat{\gamma}(\boldsymbol{\lambda})\boldsymbol{1}^\top)^\top - \hat{\boldsymbol{U}}(\boldsymbol{\lambda})\text{diag}(\hat{\boldsymbol{\sigma}}(\boldsymbol{\lambda}))\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^\top\right)_T \\ & + \lambda_0\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^\top + \epsilon\text{diag}(\hat{\boldsymbol{\sigma}}(\boldsymbol{\lambda}))\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^\top \end{aligned} \tag{A.23}$$

$$\begin{aligned} \boldsymbol{0} = & -\left(\boldsymbol{M} - \boldsymbol{X}_{I_r(\boldsymbol{\lambda})}\hat{\boldsymbol{\eta}}(\boldsymbol{\lambda})\boldsymbol{1}^\top - (\boldsymbol{Z}_{I_c(\boldsymbol{\lambda})}\hat{\gamma}(\boldsymbol{\lambda})\boldsymbol{1}^\top)^\top - \hat{\boldsymbol{U}}(\boldsymbol{\lambda})\text{diag}(\hat{\boldsymbol{\sigma}}(\boldsymbol{\lambda}))\hat{\boldsymbol{V}}(\boldsymbol{\lambda})^\top\right)_T\hat{\boldsymbol{V}}(\boldsymbol{\lambda}) \\ & + \lambda_0\hat{\boldsymbol{U}}(\boldsymbol{\lambda}) + \epsilon\hat{\boldsymbol{U}}(\boldsymbol{\lambda})\text{diag}(\hat{\boldsymbol{\sigma}}(\boldsymbol{\lambda})) \end{aligned} \tag{A.24}$$

where $(\cdot)_T$ zeroes out matrix elements that are not observed in the training set. The gradient optimality conditions with respect to $\boldsymbol{\eta}$ and $\boldsymbol{\gamma}$ are derived using the usual procedure. To get the partial derivatives of the fitted values with respect to $\boldsymbol{\lambda}$, we implicitly differentiate the gradient optimality conditions, as well as (A.21), with respect to $\boldsymbol{\lambda}$ and solve the resulting system of linear equations. The gradient of the validation loss with respect to the penalty parameters is straightforward to calculate once the partial derivatives are obtained. However, we omit this tedious calculation.

We now show that the conditions in Theorem 1 are satisfied.

Condition 1: We hypothesize that the differentiable space $S_{\boldsymbol{\lambda}}$ defined in (31) is also a local optimality space $\boldsymbol{\lambda}$. For the group lasso penalties, we use the same reasons mentioned in A.3.2 to justify this hypothesis. For the nuclear norm penalty, it has been observed empirically that small perturbations in the penalty parameter result in matrices with similar rank (Mazumder et al. 2010). This supports our belief that $S_{\boldsymbol{\lambda},\boldsymbol{\Gamma}}$ is a local optimality space with respect to $\boldsymbol{\Gamma}$ at $\boldsymbol{\lambda}$.

Condition 2: The only non-smooth components of the training criterion are the group lasso and nuclear norm penalties. The group lasso penalty is twice-differentiable when restricted to the differentiable space, using the same reasoning in Section A.3.2. From (A.18), we see that the nuclear norm $\|\boldsymbol{\Gamma}\|_*$ is also twice-differentiable with respect to $\boldsymbol{\Gamma}$ when restricted to $S_{\boldsymbol{\lambda},\boldsymbol{\Gamma}}$. ✓

Condition 3: The differentiable space for the training criterion with respect to $\boldsymbol{\Gamma}$ is a linear space. Therefore there exists some orthonormal basis of the differentiable space. Since the training criterion is the sum of convex functions with ridge penalties on all the variables, the Hessian of the training criterion is positive definite for any $\epsilon > 0$. ✓

A.4 Gradient Descent Details

Here we discuss our choice of step size and convergence criterion in gradient descent.

There are many possible choices for our step size sequence $\{t^{(k)}\}$ (Boyd & Vandenberghe 2004). We chose a backtracking line, which we describe here briefly. 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. The algorithm is given below. It depends on constants $\alpha \in (0, 0.5)$ and $\beta \in (0, 1)$. In our examples initial step size was 1, and we backtrack with parameters $\alpha = 0.001$ and $\beta = 0.1$. During gradient descent, it is possible that the step size will result in a negative regularization parameter; we reject any step that would set a regularization parameter to below a minimum threshold of 1e-6.

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 $t := \beta t$
```

Our convergence criterion is based on the change in our validation loss between iterates. More specifically, we stop our algorithm when

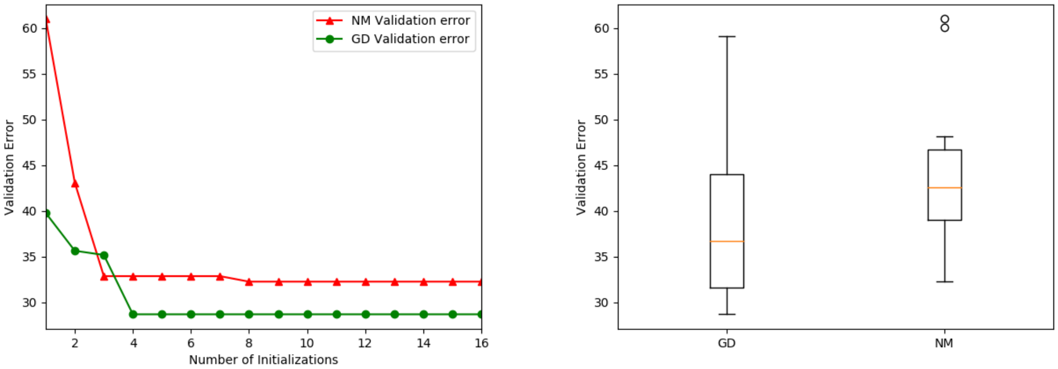
$$L\left(\boldsymbol{y}_V, f_{\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}^{(k+1)})}(\boldsymbol{X}_V)\right) - L\left(\boldsymbol{y}_V, f_{\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}^{(k)})}(\boldsymbol{X}_V)\right) \leq \delta$$

For the results in this manuscript we use $\delta = 0.0005$.

A.5 Sensitivity to initialization points

Since the results of gradient descent and Nelder-Mead depend on their initialization points, we ran a simulation to see how sensitive the methods were to where they were initialized and how many initializations were used.

Figure A.1: Error of additive models tuned by Gradient Descent vs. Nelder-Mead. Left: Validation error of models after as the number of initialization points increases. Right: The distribution of validation errors. (Gradient Descent = GD, Nelder-Mead = NM)



We tested a smaller version of the joint optimization problem in Section 2.4.2. Here we use 60 training, 30 validation, and 30 test observations and $p = 15$ covariates. The response was generated from (34). We initialized $\boldsymbol{\lambda}$ by considering all possible combinations of $(\lambda_0, \lambda_1 \mathbf{1})$ where $\lambda_0, \lambda_1 \in \{10^i : i \in \{-2, -1, 0, 1\}\}$.

In Figure A.1 (left), we plot the validation error as the number of initializations increases. The validation errors from both methods plateau quickly. Gradient descent manages to find penalty parameters with lower validation error than Nelder-Mead. Figure A.1 (right) presents the distribution of validation errors resulting from the random initializations. On average, gradient descent finds penalty parameters with lower validation error compared to Nelder-Mead. The plots show that the methods are indeed sensitive to their initialization points. For example, one could run a very coarse grid search on the two-parameter version of the joint optimization problem and use the best penalty parameter values.

A.6 Additional simulation results

The simulation results in Section 3 show that joint optimization problems with many penalty parameters can produce better models than those with only two penalty parameters. One may wonder if this difference is due to the method used to tune the penalty parameters. Here we present results from tuning the two-penalty-parameter joint optimization problems from Sections 3.2, 3.3, and 3.4 using gradient descent, Nelder-Mead, and Spearmint. As shown in Table A.1, the performance of these methods are very similar to grid search. Regardless of the method used to tune the two-penalty parameter joint optimization, the resulting models all have higher validation and test error compared to the models from the joint optimization problem with many penalty parameters tuned by gradient descent.

Table A.1: Two-parameter joint optimization problems for the examples in Section 3. Standard errors are given in parentheses. We abbreviated the methods as follows: Gradient Descent = GD, Nelder-Mead = NM, Spearmint = SP, Grid Search = GS

Sparse additive models			
	Validation Error	Test Error	# Solves
GD	23.87 (0.97)	26.10 (0.86)	13.07
NM	28.86 (1.04)	29.97 (0.96)	100
SP	29.18 (1.07)	30.09 (1.08)	100
GS	28.71 (0.97)	29.42 (0.96)	100

Sparse Group Lasso			
n=90, p=600, M=30			
	Validation Err	Test Err	# Solves
GD	46.82 (2.21)	49.33 (1.36)	21.43
NM	46.37 (2.24)	48.95 (1.35)	100
SP	45.70 (2.32)	49.35 (1.56)	100
GS	47.23 (2.26)	50.01 (1.40)	100
n=90, p=900, M=60			
	Validation Error	Test Error	# Solves
GD	45.71 (2.26)	50.31 (1.93)	20.77
NM	44.95 (2.24)	50.18 (1.82)	100
SP	49.59 (2.27)	56.54 (2.14)	100
GS	45.70 (2.27)	51.34 (1.86)	100
n=90, p=1200, M=100			
	Validation Error	Test Error	# Solves
GD	50.46 (2.30)	57.02 (1.94)	19.80
NM	49.92 (2.33)	55.46 (1.89)	100
SP	49.70 (2.26)	56.51 (2.16)	100
GS	50.00 (2.16)	57.14 (2.18)	100

Low-rank Matrix Completion			
	Validation Err	Test Err	Num Solves
GD	0.70 (0.04)	0.71 (0.04)	8.03 (0.79)
NM	0.71 (0.04)	0.71 (0.04)	100
SP	0.73 (0.04)	0.74 (0.04)	100
GS	0.71 (0.04)	0.72 (0.04)	100

A.7 Smoothness of the Validation Loss

Since our algorithm depends on the validation loss being smooth almost everywhere, a potential concern is that the validation loss may not be differentiable at the solution of the joint optimization problem. We address this concern empirically. Based on the simulation study below, we suspect that the minimizer falls exactly at a knot (where our validation loss is not differentiable with respect to $\boldsymbol{\lambda}$) with measure zero.

In this simulation we solved a penalized least squares problem with a lasso penalty and tuned the penalty parameter to minimize the loss on a separate validation set. We considered a linear model with 100 covariates. The training and validation sets included 40 and 30 observations, respectively. The response was generated data from the model

$$y = \boldsymbol{X}\boldsymbol{\beta} + \sigma\epsilon$$

where $\boldsymbol{\beta} = (1, 1, 1, 0, ..., 0)$. ϵ and $\boldsymbol{X}$ were drawn independently from a standard Gaussian distribution. σ was chosen so that the signal to noise ratio was 2. For a given $\lambda > 0$ our fitted $\boldsymbol{\beta}$ minimized the penalized training criterion

$$\hat{\boldsymbol{\beta}}(\lambda) = \arg\min_{\boldsymbol{\beta}} \|\boldsymbol{y} - \boldsymbol{X}\boldsymbol{\beta}\|_T^2 + \lambda\|\boldsymbol{\beta}\|_1$$

We then chose the λ-value for which $\hat{\boldsymbol{\beta}}(\lambda)$ minimized the validation error.

In our 500 simulation runs, the penalty parameter that minimized the validation loss was never located at a knot: Using a homotopy solver for the lasso, we were able to find the *exact* knots (λ-values where variables enter/leave the model), and these points never achieved the minimum value of the validation loss. While this is only one example, and not definitive proof, we believe it is a strong indication that it is unlikely for solutions to occur regularly at knots in penalized problems.

In addition, we believe that the behavior of our procedure is analogous to solving the Lasso via sub-gradient descent. In the Lasso setting, sub-gradient descent with a properly chosen step-size will converge to the solution. In addition, if initialized at a differentiable β-value (ie. with all non-zero entries), then the lasso objective will be differentiable at all iterates in this procedure with probability one. Admittedly, using the sub-gradient method to solve the lasso has fallen out of favor. The current gold-standard methods, such as generalized gradient descent, give sparse solutions at large enough iterates and achieve faster convergence rates.

References

Boyd, S. & Vandenberghe, L. (2004), *Convex optimization*, Cambridge university press.

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

Rellich, F. (1969), *Perturbation theory of eigenvalue problems*, CRC Press.

Tibshirani, R. J., Taylor, J. et al. (2011), 'The solution path of the generalized lasso', *The Annals of Statistics* **39**(3), 1335–1371.

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

Watson, G. A. (1992), 'Characterization of the subdifferential of some matrix norms', *Linear algebra and its applications* **170**, 33–45.

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.