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:

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

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. $\square$

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 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

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

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

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

Note that the Hessian is positive definite for any $\epsilon > 0$. Following the logic in Appendix 1.3.1, all three conditions in Theorem 1 are satisfied.

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 Hessian is positive definite for any $\epsilon > 0$. Following the logic in Appendix 1.3.1, all conditions in Theorem 1 are satisfied.

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

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

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

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