

Oracle Inequalities for multiple penalty parameters

Jean Feng*

Department of Biostatistics, University of Washington
and

Noah Simon

Department of Biostatistics, University of Washington

August 29, 2016

Abstract

In penalized least squares problems, penalty parameters determine the tradeoff between minimizing the residual sum of squares and the model complexity. The oracle set of penalty parameter values that minimize the generalization error are usually estimated by evaluating the models on a separate validation set or by cross-validation. We find that in many problems, the fitted models are smoothly parameterized by the penalty parameters. Thus we can show that the difference between the prediction error of the selected model and the oracle converges at a near-parametric rate. This finding justifies recent work on combining penalty functions using separate penalty parameters.

Keywords: Regression, Cross-validation, Regularization

*Jean Feng was supported by NIH grants ???. Noah Simon was supported by NIH grant DP5OD019820. The content is solely the responsibility of the authors and does not necessarily represent the official views of the National Institutes of Health.

1 Introduction

Per the usual regression framework, we observe response $y \in \mathbb{R}$ and predictors $\mathbf{x} \in \mathbb{R}^p$. Suppose y is generated from the true model g^* from model class \mathcal{G}

$$y = g^*(\mathbf{x}) + \epsilon \quad (1)$$

where ϵ_i are random errors. Our goal is to find the best model in \mathcal{G} to model y given \mathbf{x} .

In high-dimensional ($p \gg n$) or ill-posed problems, the ordinary least squares estimate performs poorly as it overfits to the training data. A common solution is to add regularization, or penalization, to control model complexity and induce desired structure. The penalized least squares estimate minimizes a criterion of the form

$$\hat{g}(\cdot | \boldsymbol{\lambda}) = \arg \min_{g \in \mathcal{G}} \frac{1}{n} \sum_{i=1}^n (y_i - g(x_i))^2 + \sum_{j=1}^J \lambda_j P_j(g) \quad (2)$$

where P_j are the penalty functions and λ_j are the penalty parameters.

Selecting the penalty parameters is an important task since they ultimately determine the fitted model. Their oracle values balance the residual least squares and the penalty terms to ensure fast convergence rates (Van de geer-book, Wahba-smoothing spline paper, and others?). For example, when fitting an additive model $f(\mathbf{x}) = \sum_{j=1}^J f_j(x_j)$ with a roughness penalty for each component, the penalty parameters should be inversely proportional to the penalties of the true model (cite Vandegeer additive models). When fitting a linear model using the lasso, the penalty parameter should be on the order $\sigma(\log p/n)^{1/2}$ where σ^2 is the variance of the error terms.

The obvious problem is that the oracle penalty parameters depend on unknown values. Instead the penalty parameters are usually tuned via a training/validation split or cross-validation. The basic idea is to train a model on a random partition of the data and evaluate its error on the remaining data. One then searches for the penalty parameters that yield the lowest validation error. For a more complete review of cross-validation, refer to Arlot (CITE).

The performance of cross-validation-like procedures is characterized by bounding the prediction error. Typically the upper bound is composed of two terms: the error of the oracle plus a complexity term. In a general CV framework, Van Der Laan (2003, 2004) provides finite sample oracle inequalities assuming that CV is performed over a finite model class and

Mitchell () uses an entropy approach to bound CV for potentially infinite model classes. In the regression setting, Györfi (2002) provides a finite sample inequality for training/validation split for least squares and Wegkamp (2003) proves an oracle inequality for a penalized least squares holdout procedure. There are also bounds for cross-validated models from ridge regression and lasso (Golub, Heath and Wahba, Chetverikov, and Chatterjee), though the proofs usually rely on the linearity of the model class and are therefore hard to generalize.

Despite the wealth of literature on cross-validation, there is very little work on characterizing the prediction error when the regularization method has multiple penalty parameters. A potential reason is that tuning multiple penalty parameters is computationally difficult so most regularization methods only have one or two tuning parameters (e.g. Elastic Net, Sparse Group Lasso, etc.). However, recent efforts have used continuous optimization methods to make this “hyperparameter selection” problem computationally tractable. For penalized regression problems, the gradient of the validation loss with respect to the penalty parameters can often be calculated by implicit differentiation (Bengio, Foo). Thus a gradient descent procedure can be used to tune the penalty parameters. For more general machine learning problems, one can use a gradient-free approach such as Bayesian optimization (Snoek).

Our paper provides a finite-sample upper bound on the prediction error when tuning multiple penalty parameters via a training/validation split or cross-validation. We show that if the fitted functions vary smoothly in the penalty parameters, then the error of the model from the selected penalty parameters converges to the error of the model for the oracle penalty parameters at a near-parametric rate. We find that this smoothness assumption is satisfied by many penalized regression problems. Our inequality provides a guide for the rate to increase the number of penalty parameters with the number of samples to achieve an optimal convergence rate. The proofs use results from empirical process theory and an implicit differentiation trick.

Section 2 provides bounds on the prediction error for a training/validation framework and cross-validation. Section 3 proves that for many penalized regression problems, the fitted models are smoothly parameterized by the penalty parameters. Section 4 provides simulation studies to support the theory. Section 5 discusses the results in the paper. Section 6 contains the full proofs for the main results and additional lemmas.

2 Main Result

2.1 Training/Validation Split

Given the total observed dataset D of size n , suppose it is randomly split into a training set T of size n_T and validation set V of size n_V . For a function h , define $\|h\|_V^2 = \frac{1}{n_V} \sum_{i \in V} h^2(x_i)$ and similarly for T . Using this notation, the fitted model defined in (2) can be reformulated as

$$\hat{g}(\cdot|\boldsymbol{\lambda}) = \arg \min_{g \in \mathcal{G}} \frac{1}{2} \|y - g\|_T^2 + \sum_{j=1}^J \lambda_j P_j(g) \quad (3)$$

In the training/validation framework, the selected penalty parameters minimize the validation error as follows

$$\hat{\boldsymbol{\lambda}} = \arg \min_{\boldsymbol{\lambda} \in \Lambda} \frac{1}{2} \|y - \hat{g}(\cdot|\boldsymbol{\lambda})\|_V^2 \quad (4)$$

where Λ is the range of penalty parameter values. We are interested in bounding the difference between the fitted model and the true model at the observed covariates in the validation set, which can be expressed as $\|\hat{g}(\cdot|\hat{\boldsymbol{\lambda}}) - g^*\|_V$.

Our bound is based on the basic inequality (cite?). Let the oracle penalty parameters $\tilde{\boldsymbol{\lambda}}$ be defined as

$$\tilde{\boldsymbol{\lambda}} = \arg \min_{\boldsymbol{\lambda} \in \Lambda} \frac{1}{2} \|\hat{g}(\cdot|\tilde{\boldsymbol{\lambda}}) - g^*\|^2 \quad (5)$$

where $\|h\| = \int h^2(x) d\mu(x)$. Let the set of fitted models be denoted

$$\mathcal{G}(T) = \{\hat{g}(\cdot|\boldsymbol{\lambda}) : \boldsymbol{\lambda} \in \Lambda\} \quad (6)$$

From the definition of $\hat{\boldsymbol{\lambda}}$, we have

$$\|\hat{g}(\cdot|\hat{\boldsymbol{\lambda}}) - g^*\|_V^2 \leq \|\hat{g}(\cdot|\tilde{\boldsymbol{\lambda}}) - g^*\|_V^2 + 2 \left\langle \epsilon, \hat{g}(\cdot|\hat{\boldsymbol{\lambda}}) - \hat{g}(\cdot|\tilde{\boldsymbol{\lambda}}) \right\rangle_V \quad (7)$$

$$\leq \|\hat{g}(\cdot|\tilde{\boldsymbol{\lambda}}) - g^*\|_V^2 + \sup_{g \in \mathcal{G}(T)} 2 \left\langle \epsilon, g - \hat{g}(\cdot|\tilde{\boldsymbol{\lambda}}) \right\rangle_V \quad (8)$$

where $\langle h, \ell \rangle_A = \frac{1}{\text{size}(A)} \sum_{i \in A} h(x_i) \ell(x_i)$. This upper bound can be understood as a bias-variance tradeoff. The first term is the difference between the oracle model and the truth, so this is just the bias from using the model class $\mathcal{G}(T)$. The second term is an empirical process term that can be bounded by its variance, so this can be thought of as the “variance” of $\mathcal{G}(T)$.

Empirical process theory provides a more formal framework for characterizing the behavior of the empirical process term. “Variance” is usually referred to as complexity and can be measured in a number of ways; we will use metric entropy in this paper. A more thorough review of empirical process theory is presented in Section 6. Here we only highlight the fact that functions smoothly parameterized by $\boldsymbol{\theta}$ as follows

$$\|\boldsymbol{\theta} - \boldsymbol{\theta}'\|_2 \leq d \implies \|f(\cdot|\boldsymbol{\theta}) - f(\cdot|\boldsymbol{\theta}')\|_\infty \leq O_p(d) \quad (9)$$

have low metric entropy. Hence their empirical process terms are small with high probability.

In the penalized regression setting, the fitted functions $\hat{g}(\cdot|\boldsymbol{\lambda})$ are parameterized by $\boldsymbol{\lambda}$. Moreover, we show in Section 3 that $\hat{g}(\cdot|\boldsymbol{\lambda})$ is smoothly parameterized by $\boldsymbol{\lambda}$ for many penalized regression problems. This fact combined with (7) provides a finite-sample upper bound on the error of the fitted model $\hat{g}_{\hat{\lambda}}(\cdot|T)$ over the validation points.

Theorem 1. *Suppose that $\sup_{g \in \mathcal{G}(\cdot|T)} \|g\|_\infty \leq G < \infty$. Suppose that $\Lambda = [n_V^{-t_{\min}}, n_V^{t_{\max}}]^J$.*

Suppose that if $\|\epsilon\|_T \leq 2\sigma$, there is some constant C, κ such that for any $u > 0$, we have

$$\|\boldsymbol{\lambda}_1 - \boldsymbol{\lambda}_2\| \leq Cn^\kappa u \implies \|\hat{g}_{\boldsymbol{\lambda}_1} - \hat{g}_{\boldsymbol{\lambda}_2}\|_V \leq u \quad (10)$$

Then with high probability, we have for constants c_1, c_2

$$\|\hat{g}_{\hat{\lambda}} - g^*\|_V \leq \|\hat{g}_{\hat{\lambda}} - g^*\|_V + c_1 \left(\frac{J(\log n_V + c_2)}{n_V} \right)^{1/2} + \sqrt{c_1 \left(\frac{J(\log n_V + c_2)}{n_V} \right)^{1/2} \|\hat{g}_{\hat{\lambda}} - g^*\|_V} \quad (11)$$

Theorem 1 is a special case of Theorem 3 and their proofs are given in Section 6.

The $\log n_V$ terms are the result of increasing the range of Λ at a polynomial rate. Increasing the range of Λ is important to ensure fast convergence rates.

2.2 Cross-Validation

In practice, K -fold cross-validation is a far more common procedure than a training/validation split. In addition, we will now bound the generalization error rather than the prediction error over the validation observations. Toward this end, we will apply the oracle inequality in Mitchell (CITE) to the problem of penalized regression.

The problem setup for K -fold CV is as follows. Let the K partitions for $k = 1, \dots, K$ be denoted D_k (with size n_k) and the entire set minus the D_k will be denoted D_{-k} . Consider the joint optimization problem for K -fold CV:

$$\hat{\lambda} = \arg \min_{\lambda \in \Lambda} \frac{1}{2K} \sum_{k=1}^K \|y - \hat{g}_\lambda(\cdot | D_{-k})\|_{D_k}^2 \quad (12)$$

$$\hat{g}(\cdot | \lambda, D_{-k}) = \arg \min_{g \in \mathcal{G}} \frac{1}{2} \|y - g\|_{D_{-k}}^2 + \sum_{j=1}^J \lambda_j P_j(g) \quad (13)$$

In traditional cross-validation, the final model is retrained on all the data with $\hat{\lambda}$. However, bounding its generalization error requires additional regularity assumptions (CITE mitchell). We consider the following “averaged version of cross-validation” instead

$$\hat{g}_{ACV}(\cdot) = \frac{1}{K} \sum_{k=1}^K \hat{g}(\cdot | \hat{\lambda}, D_{-k}) \quad (14)$$

The following theorem bounds the generalization error of \hat{g}_{ACV} .

Theorem 2. *Suppose the errors have expectation zero and $\|\epsilon\|_\infty < \infty$.*

Suppose $\sup_{g \in \mathcal{G}} \|g\|_\infty \leq G$.

Suppose there is a constant C such that

$$\|\hat{g}_{\lambda_1} - \hat{g}_{\lambda_2}\|_\infty \leq \|\lambda_1 - \lambda_2\| C n^\kappa \quad (15)$$

Suppose that $\Lambda = [n^{-t_{\min}}, n^{t_{\max}}]^J$.

With high probability, we have for any $a > 0$,

$$E_D \|\hat{g}_{ACV} - g^*\|^2 \leq (1 + a) \min_{k \in 1:K, \lambda \in \Lambda} E_D \|\hat{g}(\cdot | \lambda, D_{-k}) - g^*\|^2 + c_a \max_{k=1:K} \frac{\log^2(n)}{n_k} \quad (16)$$

where c_a is given in Mitchell.

Theorem 2 is a stronger result than Theorem 1, but one is required to show that \hat{g}_λ is smoothly parameterized by λ over the entire domain, not just the validation points.

2.2.1 Example

We present a nonparametric additive model as an example.

Consider the problem

$$\|y - (f + g)\|^2 + \lambda_1 \int |f^{(s)}(x)|_n^2 dx + \lambda_2 \int |g^{(t)}(x)|^2 dx \quad (17)$$

According to Vandegeer (CITE), the oracle rate is bounded above by

$$\|\hat{f}_{\hat{\lambda}} - f^*\| = O_p(n^{-\frac{s}{2s+1}}), \|\hat{g}_{\hat{\lambda}} - g^*\| = O_p(n^{-\frac{t}{2t+1}}) \quad (18)$$

In the training/validation framework, Theorem 1 gives us

$$\begin{aligned} \|\hat{f}_{\hat{\lambda}} + \hat{g}_{\hat{\lambda}} - (f^* + g^*)\|_V &= O_p(n_T^{-\frac{s}{2s+1}}) + O_p(n_T^{-\frac{t}{2t+1}}) + c_1 \left(\frac{J(\log n_V + c_2)}{n_V} \right)^{1/2} \\ &\quad + \sqrt{c_1 \left(\frac{J(\log n_V + c_2)}{n_V} \right)^{1/2} \left(O_p(n_T^{-\frac{k}{2k+1}}) + O_p(n_T^{-\frac{m}{2m+1}}) \right)} \end{aligned}$$

In the cross-validation framework, Theorem 2 gives us

$$E_D \|\hat{f}_{ACV} + \hat{g}_{ACV} - (f^* + g^*)\|^2 = (1 + a) E_D \left(O_p(n_k^{-\frac{s}{2s+1}}) + O_p(n_k^{-\frac{t}{2t+1}}) \right) + c_a \max_{k=1:K} \frac{\log^2(n)}{n_k}$$

2.2.2 Implications

Theorem 1 and 2 imply that $\hat{g}_{\hat{\lambda}}$ has a semi-parametric convergence rate: the nonparametric (or potentially parametric) convergence rate of the oracle and the parametric convergence rate of the cross-validated model to the oracle. As long as the number of penalty parameters is finite, it does not affect the convergence rate of the model asymptotically.

We can minimize the prediction error by balancing the two terms in the upper bound. The inequalities suggest that there are many approaches. One could change the ratio between the training and validation sets as the number of samples increases. Alternatively, one could increase the number of penalties and penalty parameters with the number of samples. Of course, finding the true minimizer of the upper bound will require knowing properties about the true model, so this would have to be done in some heuristic manner.

3 Smoothness of \hat{g}_{λ} in λ

We now show that \hat{g}_{λ} is smoothly parametrized by λ . Theorem 1 requires this smoothness assumption to hold over the validation observations whereas Theorem 2 requires this to hold

over the entire domain. Smoothness over the validation set is generally easier to show. We will prove it for nonparametric additive models with smooth penalties and certain nonsmooth penalties. Smoothness over the entire domain is harder to show, so we consider two specific examples: parametric regression problems (where p can grow with n) and smoothing splines.

Throughout, we will presume that \mathcal{G} is a convex function class.

3.1 Smoothness over the Validation Set

We will show that $\hat{g}(\cdot|\lambda)$ varies smoothly with respect to λ over the observed covariates, which will directly imply smoothness over the validation set. Suppose we are in the additive model setting. The fitted models minimize the training criterion

$$\{\hat{g}_j(\cdot|\lambda)\}_{j=1}^J = \arg \min_{g \in \mathcal{G}} \|\mathbf{y} - \sum_{j=1}^J g_j(\mathbf{x}_j)\|_T^2 + \sum_{j=1}^J \lambda_j P_j(g_j) \quad (19)$$

We will not directly consider functions that minimize (19). Instead we will characterize the function class that minimize the perturbed training criterion

$$\{\hat{g}_j(\cdot|\lambda)\}_{j=1}^J = \arg \min_{g \in \mathcal{G}} \|\mathbf{y} - \sum_{j=1}^J g_j(\mathbf{x}_j)\|_T^2 + \sum_{j=1}^J \lambda_j \left(P_j(g_j) + \frac{w}{2} \|g_j\|_D^2 \right) \quad (20)$$

where $w > 0$ is some constant. The additional ridge penalty will be used in the proofs to ensure that the model is “well-conditioned” over the observed covariates.

In practice, one can choose w to be small enough such that the fitted models from (20) are indistinguishable from those in (19). Lemma 5 proves that the convergence rate of the oracle penalty parameters is preserved for small enough w .

3.1.1 Smooth Penalties

We will first consider the simple case where the penalties P_j are twice-differentiable everywhere. The following lemma shows that fitted function values vary smoothly in the penalty parameters.

Lemma 1. *We suppose the penalty functions P_j are convex and twice-differentiable. Suppose that $\sup_{g \in \mathcal{G}} \|g\|_D \leq G$. Suppose $\lambda_j \geq \lambda_{\min}$ for all j .*

For all $d > 0$, any $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}$ that satisfy

$$\|\boldsymbol{\lambda}^{(1)} - \boldsymbol{\lambda}^{(2)}\| \leq \frac{dw}{2J} \left(\frac{n}{n_T \lambda_{\min}} (2G + \|\epsilon\|_T) + wG + G \right)^{-1} \lambda_{\min}$$

we have

$$\left\| \sum_{j=1}^J \hat{g}_j(\cdot | \boldsymbol{\lambda}^{(1)}) - \hat{g}_j(\cdot | \boldsymbol{\lambda}^{(2)}) \right\|_D \leq d$$

All of the proofs on the smoothness of $\hat{g}(\cdot | \boldsymbol{\lambda})$ follow the same recipe. The first step is to consider the optimization problem (19) restricted to models on the line

$$\left\{ \hat{g}(\cdot | \boldsymbol{\lambda}^{(1)}) + m \left(\hat{g}(\cdot | \boldsymbol{\lambda}^{(2)}) - \hat{g}(\cdot | \boldsymbol{\lambda}^{(1)}) \right) : m \in [0, 1] \right\} \quad (21)$$

By implicit differentiation of the KKT conditions, we can then determine how the fitted models change with respect to the penalty parameters. Finally, the difference $\|\hat{g}_{\boldsymbol{\lambda}^{(1)}} - \hat{g}_{\boldsymbol{\lambda}^{(2)}}\|$ is bounded using the mean value theorem. Depending on the situation, the result can be proven directly or by contradiction.

For illustration, we present the proof for Lemma 1 in the case where there is only one penalty parameter. The case with multiple penalty parameters is given in Section 6.

Proof of Lemma 1. Let $h = \hat{g}(\cdot | \boldsymbol{\lambda}^{(1)}) - \hat{g}(\cdot | \boldsymbol{\lambda}^{(2)})$. Suppose for contradiction that $\|h\|_D > d$. Consider the one-dimensional optimization problem

$$\hat{m}(\lambda) = \arg \min_m \frac{1}{2} \|y - (g + mh)\|_T^2 + \lambda \left(P(g + mh) + \frac{w}{2} \|g + mh\|_D^2 \right)$$

Now by the KKT conditions, we have

$$\langle y - (g + mh), h \rangle_T + \lambda \frac{\partial}{\partial m} P(g + mh) + \lambda w \langle h, g + mh \rangle_D = 0$$

It's implicit derivative with respect to λ is

$$\frac{\partial \hat{m}(\lambda)}{\partial \lambda} = \left(\|h\|_T^2 + \lambda \frac{\partial^2}{\partial m^2} P(g + mh) + \lambda w \|h\|_D^2 \right)^{-1} \left(\frac{\partial}{\partial m} P(g + mh) + w \langle h, g + mh \rangle_D \right) \quad (22)$$

From the KKT conditions, we can show

$$\left| \frac{\partial}{\partial m} P(g + mh) \right| \leq \left(\frac{n}{n_T \lambda_{\min}} (2G + \|\epsilon\|_T) + wG + G \right) \|h\|_D$$

Hence

$$\left| \frac{\partial}{\partial \lambda} \hat{m}(\lambda) \right| \leq \left(\frac{n}{n_T} n^{\tau_{\min}} (2G + \|\epsilon\|_T) + wG + G \right) n^{\tau_{\min}} w^{-1} \|h\|_D^{-1}$$

By the MVT, there is some $\alpha \in (\lambda^{(1)}, \lambda^{(2)})$ such that

$$\begin{aligned} |\hat{m}(\lambda^{(2)}) - \hat{m}(\lambda^{(1)})| &= \left| (\lambda^{(2)} - \lambda^{(1)}) \frac{\partial \hat{m}(\lambda)}{\partial \lambda} \right|_{\lambda=\alpha} \\ &\leq |\lambda^{(2)} - \lambda^{(1)}| \left(\frac{n}{n_T \lambda_{\min}} (2G + \|\epsilon\|_T) + wG + G \right) \frac{1}{w d \lambda_{\min}} \\ &= 1/2 \end{aligned}$$

But this is a contradiction since we know that $\hat{m}(\lambda^{(2)}) = 1$ and $\hat{m}_{\tilde{k}}(\lambda^{(1)}) = 0$. \square

3.1.2 Nonsmooth penalties

If the regression problem contains non-smooth penalty functions, similar results do not necessarily hold. Nonetheless, we find that for many popular non-smooth penalty functions, the functions $\hat{g}(\cdot | \boldsymbol{\lambda})$ are still smoothly parameterized by $\boldsymbol{\lambda}$ almost everywhere. To characterize such problems, we use the approach in Feng & Simon (CITE). We begin with the following definitions:

Definition 1. *The differentiable space of a real-valued function L at $g \in \mathcal{G}$ is the set of functions*

$$\Omega^L(g) = \left\{ h \in \mathcal{G} \left| \lim_{\epsilon \rightarrow 0} \frac{L(g + \epsilon h) - L(g)}{\epsilon} \text{ exists} \right. \right\} \quad (23)$$

Definition 2. *S is a local optimality space for a convex function $L(\cdot, \boldsymbol{\lambda}_0)$ if there exists a neighborhood W containing $\boldsymbol{\lambda}_0$ such that for every $\boldsymbol{\lambda} \in W$,*

$$\arg \min_{g \in \mathcal{G}} L(g, \boldsymbol{\lambda}) = \arg \min_{g \in S} L(g, \boldsymbol{\lambda}) \quad (24)$$

Let the training criterion be denoted

$$L_T(g, \lambda) = \arg \min_{g \in \mathcal{G}} \left\| \mathbf{y} - \sum_{j=1}^J g_j(\mathbf{x}_j) \right\|_T^2 + \sum_{j=1}^J \lambda_j \left(P_j(g_j) + \frac{w}{2} \|g_j\|_D^2 \right)$$

We will need following conditions to hold for almost every $\boldsymbol{\lambda}$:

Condition 1. *The differentiable space $\Omega^{L_T(\cdot, \boldsymbol{\lambda})}(\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}))$ is a local optimality space for $L_T(\cdot, \boldsymbol{\lambda})$.*

Condition 2. $L_T(\cdot, \boldsymbol{\lambda})$ is twice continuously differentiable along directions in $\Omega^{L_T(\cdot, \boldsymbol{\lambda})}(\hat{\boldsymbol{\theta}}(\boldsymbol{\lambda}))$.

Nonsmooth penalties that satisfy Conditions 1 and 2 include ? (I'm not sure which nonparametric penalties actually satisfy this.)

Equipped with the conditions above, we can characterize the smoothness of the fitted functions when the penalties are nonsmooth. In fact the result is exactly the same as Lemma 1. The proof is given in Section 6.

Lemma 2. Suppose that $\sup_{g \in \mathcal{G}} \|g\|_D \leq G$. Suppose $\lambda_j \geq \lambda_{\min}$ for all j . Suppose the penalty functions are convex. Suppose Conditions 1 and 2 hold for almost every $\boldsymbol{\lambda}$.

For all $d > 0$, any $\boldsymbol{\lambda}^{(1)}, \boldsymbol{\lambda}^{(2)}$ that satisfy

$$\|\boldsymbol{\lambda}^{(1)} - \boldsymbol{\lambda}^{(2)}\| \leq \frac{dw}{2J} \left(\frac{n}{n_T \lambda_{\min}} (2G + \|\epsilon\|_T) + wG + G \right)^{-1} \lambda_{\min}$$

we have

$$\left\| \sum_{j=1}^J \hat{g}_j(\cdot | \boldsymbol{\lambda}^{(1)}) - \hat{g}_j(\cdot | \boldsymbol{\lambda}^{(2)}) \right\|_D \leq d$$

3.2 Smoothness over the entire domain

To show that the fitted functions $\hat{g}(\cdot | \boldsymbol{\lambda})$ vary smoothly with respect to $\boldsymbol{\lambda}$ over the entire domain, we will need additional assumptions. In Section 3.1, we controlled the difference between the fitted values at the validation points by adding a ridge penalty. Unfortunately this trick does not allow us to control the sup norm between the fitted functions.

Instead we will just consider specific regression problems. In the parametric regression setting, smoothness over the entire domain is easy to control as long as the derivative of the penalty is controlled by the 2-norm of the model parameters. In the smoothing spline problem, we rely on special properties of the roughness penalty.

3.2.1 Parametric Regression

Consider the parametric regression setting where the model parameters have dimension p , where p can potentially grow with the number of samples n . Again, we consider the perturbed regression problem:

$$\hat{\theta}(\boldsymbol{\lambda}) = \arg \min_{\theta \in \Theta} \|y - g(X|\theta)\|_T^2 + \sum_{j=1}^J \lambda_j \left(P_j^{v_j}(\theta) + \frac{w}{2} \|\theta\|_2^2 \right) \quad (25)$$

where the additional ridge penalty is now over the model parameters rather than the fitted values.

We can show smoothness over the entire domain under the following conditions

Condition 3. *There exists some constant K such that $\frac{\partial}{\partial m}P(\theta + m\beta) \leq K\|\beta\|_2$*

Condition 4. *There exist constants L, r such that the functions are Lp^r -Lipschitz in the model parameters:*

$$\|g(\cdot|\theta^{(1)}) - g(\cdot|\theta^{(2)})\|_\infty \leq Lp^r\|\theta^{(1)} - \theta^{(2)}\|_2 \quad (26)$$

It is easy to show that Condition 3 is satisfied by many popular parametric penalties, such as the ridge penalty $\|\cdot\|_2^2$, lasso $\|\cdot\|_1$, and group lasso $\|\cdot\|_2$. (Proofs are given in Section 6, if you insist.) Condition 4 is also quite reasonable (I think). For example, it is satisfied in linear regression when the covariates are bounded. Given these conditions, we can show that the fitted values are smooth with respect to the penalty parameters over the entire domain.

Lemma 3. *Suppose*

$$\sup_{\theta \in \Theta} \|\theta\| \leq G$$

Suppose Condition 3 and 4 are satisfied. Then for any $d > 0$ and $\lambda^{(1)}, \lambda^{(2)} \in \Lambda$ that satisfy

$$\|\lambda^{(2)} - \lambda^{(1)}\|_2 \leq d \frac{wJ\lambda_{\min}}{2Lp^r(K + wG)}$$

we have

$$\|g(\cdot|\hat{\theta}_{\lambda^{(1)}}) - g(\cdot|\hat{\theta}_{\lambda^{(2)}})\|_\infty \leq d$$

3.2.2 Smoothing Splines with a Sobolev Penalty

Finally, we consider the additive regression model of fitting a smoothing spline using the Sobolev penalty (CITE: de Boor (1978), Wahba (1990), Green & Silverman (1994)). For a given set of penalty parameters, the smoothing spline estimate is

$$\{\hat{g}_j(\cdot|\boldsymbol{\lambda})\}_{j=1}^J = \arg \min_{g_j \in \mathcal{G}} \frac{1}{2} \|y - \sum_{j=1}^J g_j\|_T^2 + \sum_{j=1}^J \lambda_j \int (g_j^{(r_j)}(x))^2 dx \quad (27)$$

where $g^{(r)}$ is the derivative of g of order $r \geq 2$. Unlike the previous sections, we will not need an additional ridge penalty to control the model class.

Due to the special property of the Sobolev penalty given in Lemma 6, we can prove a stronger statement compared to the previous Lemmas 1, 2, and 3. The following lemma shows that the fitted models are Lipschitz in the penalty parameters.

Lemma 4. *Suppose $\sup_{g \in \mathcal{G}} \|g\|_\infty \leq G$. Suppose $\lambda_j \geq \lambda_{\min}$ for all j . Then*

$$\left\| \sum_{j=1}^J \hat{g}_j(\cdot | \lambda^{(1)}) - \hat{g}_j(\cdot | \lambda^{(2)}) \right\|_\infty \leq \|\lambda^{(1)} - \lambda^{(2)}\| \frac{G}{\lambda_{\min}} \sqrt{\frac{1}{2\lambda_{\min}} \|\epsilon\|_T^2 + \frac{\lambda_{\max}}{\lambda_{\min}} \sum_{j=1}^J P(g_j^*)} \quad (28)$$

4 Simulations

We now provide a simulation study for the prediction error bound given in Theorem 3. The penalty parameters are chosen by a training/validation split. We show that the error of the select model converges to that of the oracle model at the expected $(\log(n_V)/n_V)^{1/2}$ rate.

Observations were generated from the model

$$y = \sin(x_1) + \frac{1}{2} \sin(2x_2 + 1) + \sigma \epsilon \quad (29)$$

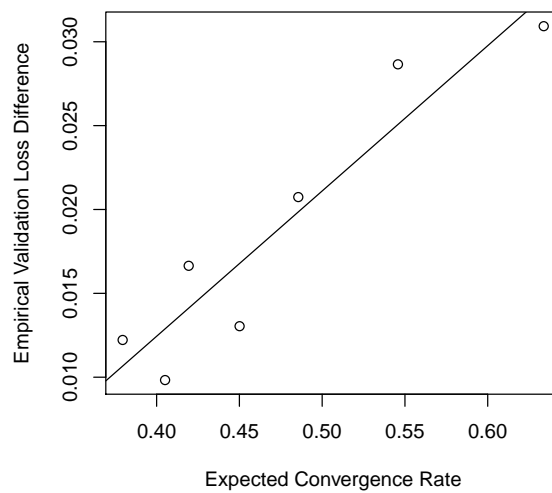
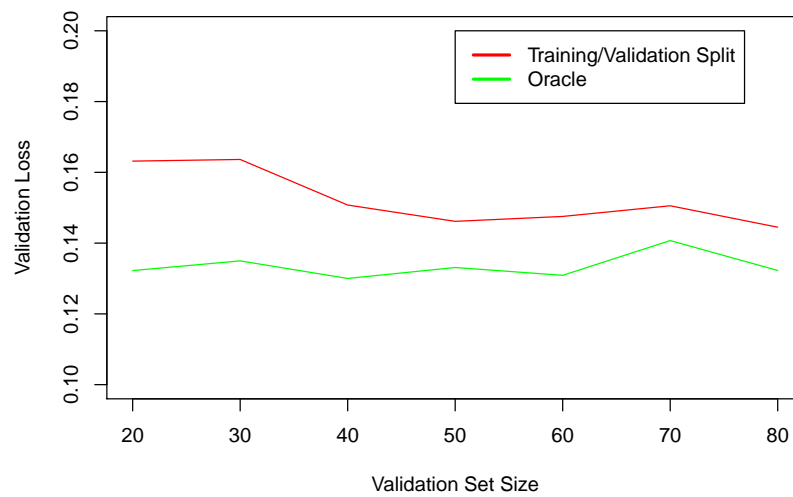
where $\epsilon \sim N(0, 1)$ and σ scaled the error term such that the signal to noise ratio was 2. The covariates x_1 and x_2 were uniformly distributed over the interval $(0, 6)$. Smoothing splines were fit with a Sobolev penalty

$$\hat{g}_{1,\lambda}, \hat{g}_{2,\lambda} = \arg \min_{g_1, g_2} \|y - f_1(x_1) - f_2(x_2)\|_T^2 + \int_0^6 (f_1^{(2)}(x))^2 dx + \int_0^6 (f_2^{(2)}(x))^2 dx \quad (30)$$

The training set contained 100 samples and models were fitted with 10 knots. A grid search was performed over the penalty parameter values $\{10^{-6+0.2i} : i = 0, \dots, 25\}$. We tested validation set sizes $n_V = 20, 30, \dots, 80$. The oracle penalty parameters were chosen by minimizing the difference between the fitted model and the true model over a separate test set of 800 samples. A total of 30 simulations were run for each validation set size.

Figure 4 plots the validation loss $\|\hat{g}_\lambda - g^*\|_V$ of the model tuned using a validation set versus the model fit using the oracle penalty parameters. As the validation set increases, the error of the tuned model converges towards the oracle model as expected. In addition we compare the observed difference between the validation losses for the two models and the expected convergence rate of $(\log(n_V)/n_V)^{1/2}$. The plot shows that theory closely matches the empirical evidence.

Figure 1: Top: Comparison between the model for penalty parameters chosen by a training/validation split vs. the oracle. Bottom: The expected versus the empirical validation loss. The line is the best fit from least squares.



5 Discussion

In this paper, we have shown that the difference in prediction error of the model chosen by cross-validation and the oracle model decreases at a near-parametric rate if the fitted models are smoothly parameterized in terms of the penalty parameters. For many penalized regression problems, we find that this is indeed the case. Our results show that adding penalty parameters does not drastically increase the model complexity. This supports recent efforts to combine regularization methods and “un-pool” regularization parameters. Furthermore, since our result holds for a search over a dense set of penalty parameters, our prediction error bounds apply to cross-validation over a continuum of values, as done in hyper-parameter optimization methods.

The main caveat is that our theorems bound the prediction error of the global minimizer of the validation set. However this is hard to achieve practically since the validation loss is not convex in the penalty parameters. More investigation needs to be done to bound the prediction error of fitted models that are local minima.

6 The Proof

In this paper, we will measure the complexity of $\mathcal{G}(T)$ by its metric entropy. Let us recall its definition here:

Definition 3. Let the covering number $N(u, \mathcal{G}, \|\cdot\|)$ be the smallest set of u -covers of \mathcal{G} with respect to the norm $\|\cdot\|$. The metric entropy of \mathcal{G} is defined as the log of the covering number:

$$H(u, \mathcal{G}, \|\cdot\|) = \log N(u, \mathcal{G}, \|\cdot\|) \quad (31)$$

Theorem 3. Let ϵ be independent sub-Gaussian random variables. Suppose that $\sup_{g \in \mathcal{G}} \|g\|_\infty \leq G < \infty$. Suppose for any training dataset $T \subseteq D$ with $\|\epsilon\|_T \leq 2\sigma$, we have

$$\int_0^R H^{1/2}(u, \mathcal{G}(\cdot|T), \|\cdot\|_V) du \leq \psi(n, J, \sigma) \quad (32)$$

Then for all $\delta > 0$ such that

$$\sqrt{n_V} \delta^2 \geq c [\psi_T(2 \|\hat{g}_\lambda - g^*\|_V + 2\delta) \vee (2 \|\hat{g}_\lambda - g^*\|_V + 2\delta)] \quad (33)$$

Then with high probability, we have

$$\|\hat{g}_{\hat{\lambda}}(\cdot|T) - g^*\|_V \leq \min_{\lambda \in \Lambda} \|\hat{g}_{\lambda}(\cdot|T) - g^*\|_V + \delta \quad (34)$$

Proof. Chaining and peeling. □

Proof of Theorem 1

Proof. By Lemma `param_covering_cube`, we have

$$H(u, \mathcal{G}(T), \|\cdot\|_V) \leq \log \frac{1}{C_J} + J \log \left(\frac{2n^{t_{max}-\kappa} + 2Cu}{Cu} \right)$$

Let $R_1 = R \wedge \sqrt{n^{t_{max}-\kappa}/C}$.

Then after immense algebraic massaging, we get

$$\int_0^R H^{1/2}(u, \mathcal{G}(T), \|\cdot\|_V) du \leq R \left(\left[\log \frac{1}{C_J} + J(2 + \log 4) + J \log \left(\frac{4n^{t_{max}-\kappa}}{C} \right) \right]^{1/2} + \sqrt{2J \log \frac{1}{R} \vee 0} \right) \quad (35)$$

We note since $\delta > \frac{1}{n_V}$ (modulo a constant), it suffices to choose δ such that

$$\sqrt{n_V} \delta^2 \geq c (\|\hat{g}_{\hat{\lambda}} - g^*\|_V + \delta) \left(\left[\log \frac{1}{C_J} + J(1 + \log 4) + J \log \left(\frac{4n^{t_{max}-\kappa}}{C} \right) \right]^{1/2} + \sqrt{J \log n_V} \right)$$

Let

$$K = c \left(\left[\log \frac{1}{C_J} + J(1 + \log 4) + J \log \left(\frac{4n^{t_{max}-\kappa}}{C} \right) \right]^{1/2} + \sqrt{J \log n_V} \right)$$

and

$$\omega = \|\hat{g}_{\hat{\lambda}} - g^*\|_V$$

The quadratic formula gives us that

$$\delta \geq \frac{K + \sqrt{K^2 + 4K\omega\sqrt{n_V}}}{2\sqrt{n_V}}$$

□

Proof of Theorem 2

Lemma 5. *The oracle rate isn't changed when we add the ridge penalty*

Proof of Lemma 1

Proof of Lemma 2

Proof of Lemma 3

Lemma 6. *Sobolev penalty has nice properties*

Proof of Lemma 4