Approximate Leave-One-Out for Fast Parameter Tuning in High Dimensions

Wenda Zhou*[1], Shuaiwen Wang*[1], Peng Xu[1], Haiho Lu[2], Arian Maleki[1], and Vahab Mirrokni[3]

[1]Columbia University, [2]M.I.T., [3]Google Research

Abstract

Consider the class of learning schemes which is composed of a sum of losses over every data point and a regularizer that imposes special structures on the model parameter and controls the model complexity. A tuning parameter, typically adjusting the amount of regularization, is necessary for this framework to work well. Finding the optimal tuning is a challenging problem in high-dimensional regimes where both the sample size and the dimension of the parameter space are large. We propose two frameworks to obtain a computationally efficient approximation of the leave-one-out cross validation (LOOCV) risk for nonsmooth losses and regularizers. Our two frameworks are based on the primal and dual formulations of the aforementioned learning scheme. We then prove the equivalence of the two approaches under smoothness conditions. This equivalence enables us to justify the accuracy of both methods under such conditions. Finally we apply our approaches to several standard problems, including generalized LASSO and support vector machines, and empirically demonstrate the effectiveness of our results.

1 Introduction

1.1 Motivation

Consider a standard prediction problem in which a dataset $\{(y_j, \boldsymbol{x}_j)\}_{j=1}^n \subset \mathbb{R} \times \mathbb{R}^p$ is employed to learn a model for inferring information about new datapoints that are yet to be observed. One of the most popular classes of learning schemes, especially in high-dimensional settings, studies the following optimization problem:

$$\hat{\boldsymbol{\beta}} := \arg\min_{\boldsymbol{\beta}} \sum_{j=1}^{n} \ell(\boldsymbol{x}_j^\top \boldsymbol{\beta}; y_j) + \lambda R(\boldsymbol{\beta}), \tag{1}$$

where $\ell : \mathbb{R}^2 \to \mathbb{R}$ is the loss function, $R : \mathbb{R}^p \to \mathbb{R}$ is the regularizer, and λ is the tuning parameter that specifies the amount of regularization. With a proper regularizer in (1), we are able to achieve better bias-variance trade-off and pursue special structures such as sparsity and low rank structure. However, the performance of such techniques hinges upon the selection of tuning parameters.

The most generally applicable tuning method is cross validation [14]. One common choice is k-fold cross validation, which however presents potential bias issues in high-dimensional settings where n is comparable to p. For instance, the phase transition phenomena that happen in such

*Equal contributions.

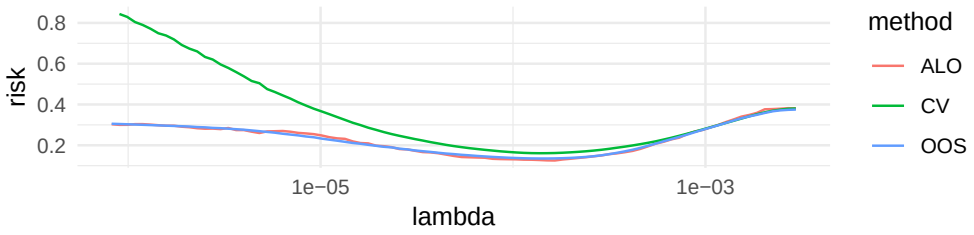

Figure 1: Risk estimates of LASSO based on 5-fold CV and ALO proposed in this paper, compared with the true out-of-sample prediction error (OOS). In this example, 5-fold CV exhibits significant bias, wherease ALO is unbiased. Here we use $n = 1000$, $p = 800$ and *iid* Gaussian design.

regimes [2] indicate that any data splitting may cause dramatic effects on the solution of (1) (see Figure 1 for an example). Hence, the risk estimates obtained from k-fold cross validation may not be reliable. The bias issues of k-fold cross validation may be alleviated by choosing the number of folds k to be large. For instance, choosing $n = k$ leads to LOOCV, which is unbiased in high-dimensional problems. However, the computation of LOOCV requires training the model n times, which is very demanding (if not impossible) for large datasets.

The high computational complexity of LOOCV has motivated researchers to propose computationally less demanding approximations of the quantity. Early examples offered approximations for the case $R(\boldsymbol{\beta}) = \frac{1}{2}\|\boldsymbol{\beta}\|_2^2$ and the loss function being smooth [1, 5, 10, 11]. In [3], the authors considered such approximations for smooth loss functions and smooth regularizers. In this line of work, the accuracy of the approximations was either not studied or was only studied in the n large, p fixed regime. In a recent paper, [12] employed a similar approximation strategy to obtain approximate leave-one-out formulas for smooth loss functions and smooth regularizers. They show that under some mild conditions, such approximations are accurate in high-dimensional settings. Unfortunately, the approximations offered in [12] only cover twice differentiable loss functions and regularizers. On the other hand, numerous modern regularizers, such as generalized LASSO and nuclear norm, and also many loss functions are not smooth.

In this paper, we propose two powerful frameworks for calculating an approximate leave-one-out estimator (ALO) of the LOOCV risk that are capable of offering accurate parameter tuning even for non-differentiable losses and regularizers. Our first approach is based on the smoothing and quadratic approximation of the primal problem (1). The second approach is based on the approximation of the dual of (1). While the two approaches consider different approximations that happen in different domains, we will show that when both ℓ and r are twice differentiable, the two frameworks produce the same ALO formulas, which are also the same as the formulas proposed in [12].

We use our platforms to obtain concise formulas for several popular examples including generalized LASSO and support vector machine (SVM). As will be clear from our examples, despite of the equivalence of the two frameworks for smooth loss functions and regularizers, the technical aspects of the derivations involved for obtaining ALO formulas have major variations in different examples. Finally, we present simulations to confirm the accuracy of our formulas on various important machine learning models. Code is available at `https://github.com/wendazhou/alocv-package`.

1.2 Notation

Lowercase and uppercase bold letters denote vectors and matrices, respectively. For subsets $A \subset \{1, 2, \ldots, n\}$ and $B \subset \{1, 2, \ldots, p\}$ of indices and a matrix $\boldsymbol{X}$, let $\boldsymbol{X}_{A,\cdot}$ and $\boldsymbol{X}_{\cdot,\boldsymbol{B}}$ denote the submatrices that include only rows of $\boldsymbol{X}$ in A, and columns of $\boldsymbol{X}$ in B respectively. Let $\{a_i\}_{i\in S}$

denote the vector whose components are a_i for $i \in S$. We may omit S, in which case we consider all indices valid in the context. For a function $f : \mathbb{R} \to \mathbb{R}$, let $\dot{f}$, $\ddot{f}$ denote its 1^{st} and 2^{nd} derivatives. For a vector $\boldsymbol{a}$, we use $\text{diag}[\boldsymbol{a}]$ to denote a diagonal matrix $\boldsymbol{A}$ with $A_{ii} = a_i$. Finally, let ∇R and $\nabla^2 R$ denote the gradient and Hessian of a function $R : \mathbb{R}^p \to \mathbb{R}$.

2 Preliminaries

2.1 Problem Description

In this paper, we study the statistical learning models in form (1). For each value of λ, we evaluate the following LOOCV risk estimate with respect to some error function d:

$$\text{loo}_\lambda := \frac{1}{n} \sum_{i=1}^{n} d(y_i, \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}^{/i}), \tag{2}$$

where $\hat{\boldsymbol{\beta}}^{/i}$ is the solution of the leave-i-out problem

$$\hat{\boldsymbol{\beta}}^{/i} := \arg\min_{\boldsymbol{\beta}} \sum_{j \neq i} \ell(\boldsymbol{x}_j^\top \boldsymbol{\beta}; y_j) + \lambda R(\boldsymbol{\beta}). \tag{3}$$

Calculating (3) requires training the model n times, which may be time-consuming in high-dimensions. As an alternative, we propose an estimator $\tilde{\boldsymbol{\beta}}^{/i}$ to approximate $\hat{\boldsymbol{\beta}}^{/i}$ based on the full-data estimator $\hat{\boldsymbol{\beta}}$ to reduce the computational complexity. We consider two frameworks for obtaining $\tilde{\boldsymbol{\beta}}^{/i}$, and denote the corresponding risk estimate by:

$$\text{alo}_\lambda := \frac{1}{n} \sum_{i=1}^{n} d(y_i, \boldsymbol{x}_i^\top \tilde{\boldsymbol{\beta}}^{/i}). \tag{4}$$

The estimates we obtain will be called approximate leave-one-out (ALO) throughout the paper.

2.2 Primal and Dual Correspondence

The objective function of penalized regression problem with loss ℓ and regularizer R is given by:

$$P(\boldsymbol{\beta}) := \sum_{j=1}^{n} \ell(\boldsymbol{x}_j^\top \boldsymbol{\beta}; y_j) + R(\boldsymbol{\beta}). \tag{5}$$

Here and subsequently, we absorb the value of λ into R to simplify the notation. We also consider the Lagrangian dual problem, which can be written in the form:

$$\min_{\boldsymbol{\theta} \in \mathbb{R}^n} D(\boldsymbol{\theta}) := \sum_{j=1}^{n} \ell^*(-\theta_j; y_j) + R^*(\boldsymbol{X}^\top \boldsymbol{\theta}), \tag{6}$$

where ℓ^* and R^* denote the *Fenchel conjugates* [1] of ℓ and R respectively. It is known that under mild conditions, (5) and (6) are equivalent [4]. In this case, we have the primal-dual correspondence relating the primal optimal $\hat{\boldsymbol{\beta}}$ and the dual optimal $\hat{\boldsymbol{\theta}}$:

$$\begin{aligned} \hat{\boldsymbol{\beta}} &\in \partial R^*(\boldsymbol{X}^\top \hat{\boldsymbol{\theta}}), \qquad & \boldsymbol{X}^\top \hat{\boldsymbol{\theta}} &\in \partial R(\hat{\boldsymbol{\beta}}), \\ \boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}} &\in \partial \ell^*(-\hat{\theta}_j; y_j), \qquad & -\hat{\theta}_j &\in \partial \ell(\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}; y_j), \end{aligned} \tag{7}$$

where ∂f denotes the set of subgradients of a function f. Below we will use both primal and dual perspectives for approximating loo_λ.

[1] The Fenchel conjugate f^* of a function f is defined as $f^*(x) := \sup_y \{\langle x, y \rangle - f(y)\}$.

3 Approximation in the Primal and Dual Domain

3.1 Approximation in the Dual Domain

We illustrate our dual method in deriving an ALO formula through a simple example of the standard LASSO. The LASSO estimator, first proposed in [15], can be formulated as the penalized regression framework in (5) by setting $\ell(\mu; y) = (\mu - y)^2/2$, and $R(\boldsymbol{\beta}) = \lambda\|\boldsymbol{\beta}\|_1$.

We recall the general formulation of the dual for penalized regression problems (6), and note that in the case of the LASSO we have:

$$\ell^*(\theta_i; y_i) = \frac{1}{2}(\theta_i - y_i)^2, \qquad R^*(\boldsymbol{\beta}) = \begin{cases} 0 & \text{if } \|\boldsymbol{\beta}\|_\infty \le \lambda, \\ +\infty & \text{otherwise.} \end{cases}$$

In particular, we note that the solution of the dual problem (6) can be obtained as:

$$\hat{\boldsymbol{\theta}} = \Pi_{\Delta_X}(\boldsymbol{y}), \qquad \text{with} \quad \Delta_X = \{\boldsymbol{\theta} \in \mathbb{R}^n : \|\boldsymbol{X}^\top \boldsymbol{\theta}\|_\infty \le \lambda\}. \tag{8}$$

where Π_{Δ_X} denotes the projection onto the polytope Δ_X. Let us now consider the leave-i-out problem. Unfortunately, the dimension of the dual problem is reduced by 1 for the leave-i-out problem, making it difficult to leverage the information from the full-data solution to approximate the leave-i-out solution. We augment the leave-i-out problem with a virtual i^{th} observation that does not affect the result of the optimization, but restores the dimensionality of the problem.

More precisely, let $\boldsymbol{y}_a$ be the same as $\boldsymbol{y}$, except that its i^{th} coordinate is replaced by $\hat{y}_i^{/i} = \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}^{/i}$, the leave-$i$-out predicted value. We note that the leave-i-out solution $\hat{\boldsymbol{\beta}}^{/i}$ is also the solution for the following augmented problem:

$$\min_{\boldsymbol{\beta} \in \mathbb{R}^p} \textstyle\sum_{j=1}^n \ell(\boldsymbol{x}_j^\top \boldsymbol{\beta}; y_{a,j}) + R(\boldsymbol{\beta}). \tag{9}$$

Let $\hat{\boldsymbol{\theta}}^{/i}$ be the corresponding dual solution of (9). Then, by (8), we know that $\hat{\boldsymbol{\theta}}^{/i} = \Pi_{\Delta_X}(\boldsymbol{y}_a)$. Additionally, the primal-dual correspondence (7) gives that $\hat{\boldsymbol{\theta}}^{/i} = \boldsymbol{y}_a - \boldsymbol{X}\hat{\boldsymbol{\beta}}^{/i}$, which is the residual in the augmented problem, and hence that $\hat{\theta}_i^{/i} = 0$. These two features allow us to characterize the leave-i-out predicted value $\hat{y}_i^{/i}$:

$$\boldsymbol{e}_i^\top \Pi_{\Delta_X}\big(\boldsymbol{y} - (y_i - \hat{y}_i^{/i})\boldsymbol{e}_i\big) = 0 \tag{10}$$

where $\boldsymbol{e}_i$ denotes the i^{th} standard vector. Solving exactly for the above equation is in general an expensive procedure that is computationally comparable to fitting the model. However, we may attempt to obtain an approximate solution of (10) by linearizing the projection operator at the full data solution $\hat{\boldsymbol{\theta}}$, or equivalently performing a single *Newton step* to solve the leave-i-out problem from the full data solution. The approximate leave-i-out fitted value $\tilde{y}_i^{/i}$ is thus given by:

$$\tilde{y}_i^{/i} = y_i - \frac{\hat{\theta}_i}{J_{ii}}, \tag{11}$$

where $\boldsymbol{J}$ denotes the Jacobian of the projection operator Π_{Δ_X} at the full data problem $\boldsymbol{y}$. Note that Δ_X is a polytope, and thus the projection onto Δ_X is almost everywhere locally affine [18]. Furthermore, it is straightforward to calculate the Jacobian of Π_{Δ_X}. Let $E = \{j : |\boldsymbol{X}_j^\top \hat{\boldsymbol{\theta}}| = \lambda\}$ be the equicorrelation set (where $\boldsymbol{X}_j$ denotes the j^{th} column of $\boldsymbol{X}$). Then the projection at the full data problem $\boldsymbol{y}$ is locally given by a projection onto the orthogonal complement of the span of $\boldsymbol{X}_{\cdot,E}$, thus yielding $\boldsymbol{J} = \boldsymbol{I} - \boldsymbol{X}_{\cdot,E}(\boldsymbol{X}_{\cdot,E}^\top \boldsymbol{X}_{\cdot,E})^{-1}\boldsymbol{X}_{\cdot,E}^\top$. We can then obtain $\tilde{y}^{/i}$ by plugging $\boldsymbol{J}$ in (11). Finally, by replacing $\boldsymbol{x}_i^\top \tilde{\boldsymbol{\beta}}^{/i}$ with $\tilde{y}_i^{/i}$ in (4) we obtain an estimate of the risk.

This approach can be extended to general loss functions and regularizers. For more information, refer to Section 3 of [20].

3.2 Approximation in the Primal Domain

In this section, we illustrate the primal approach for deriving ALO. We focus on the piecewise smooth loss functions and twice differentiable regularizers. However, the same procedure can be used for nonsmooth regularizers as well. We refer the readers to [20] for more details.

To start with, we describe the primal approach for smooth losses and regularizers. This will serve as a building block for resolving nonsmooth cases. Now to obtain loo_λ we need to solve

$$\hat{\boldsymbol{\beta}}^{/i} := \arg\min_{\boldsymbol{\beta}} \ \textstyle\sum_{j\neq i} \ell(\boldsymbol{x}_j^\top \boldsymbol{\beta}; y_j) + R(\boldsymbol{\beta}). \tag{12}$$

Assuming $\hat{\boldsymbol{\beta}}^{/i}$ is close to $\hat{\boldsymbol{\beta}}$, we can take a *Newton step* from $\hat{\boldsymbol{\beta}}$ towards $\hat{\boldsymbol{\beta}}^{/i}$ to obtain its approximation $\tilde{\boldsymbol{\beta}}^{/i}$ as:

$$\tilde{\boldsymbol{\beta}}^{/i} = \hat{\boldsymbol{\beta}} + \left[\textstyle\sum_{j\neq i} \boldsymbol{x}_j \boldsymbol{x}_j^\top \ddot{\ell}(\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}; y_j) + \nabla^2 R(\hat{\boldsymbol{\beta}})\right]^{-1} \boldsymbol{x}_i \dot{\ell}(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}; y_i). \tag{13}$$

We have by the matrix inversion lemma [7]:

$$\boldsymbol{x}_i^\top \tilde{\boldsymbol{\beta}}^{/i} = \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}} + \frac{H_{ii} \dot{\ell}(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}; y_i)}{1 - H_{ii} \ddot{\ell}(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}; y_i)}, \quad \boldsymbol{H} = \boldsymbol{X}[\boldsymbol{X}^\top \text{diag}[\{\ddot{\ell}(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}; y_i)\}_i]\boldsymbol{X} + \nabla^2 R(\hat{\boldsymbol{\beta}})]^{-1} \boldsymbol{X}^\top. \tag{14}$$

This is the formula reported in [12]. By calculating $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{H}$ in advance, we can cheaply approximate the leave-i-out prediction for all i and efficiently evaluate the LOOCV risk. However twice differentiability of both the loss and the regularizer is necessary in a neighborhood of $\hat{\boldsymbol{\beta}}$ to use the above strategy. This assumption is violated for many machine learning models including LASSO, robust regression [8], and SVM. Next we introduce a smoothing technique which lifts the scope of the above primal approach to nondifferentiable losses and regularizers. We first clarify our assumptions on the loss function.

Definition 3.1. *A singular point of a function is called q^{th} order, if at this point the function is q times differentiable, but its $(q+1)^{th}$ order derivative does not exist.*

Below we assume the loss ℓ is piecewise twice differentiable with k zero-order singularities $v_1, \ldots, v_k \in \mathbb{R}$. The existence of singularities prohibits us from directly applying strategies in (13) and (14), where twice differentiability of ℓ and R is necessary. A natural solution is to first smooth the loss ℓ, then apply the above framework for smooth objectives to the smoothed version and finally reduce the smoothness to recover the ALO formula for the original nonsmooth problem.

As the first step, consider the following smoothing idea:

$$\ell_h(\mu; y) =: \frac{1}{h} \int \ell(u; y) \phi((\mu - u)/h) du,$$

where $h > 0$ is fixed and ϕ is a smooth symmetric function which verifies: (i) *Normalization*: $\int \phi(w)dw = 1$, $\phi(w) \geq 0$, $\phi(0) > 0$; (ii) *Compact support*: $\text{supp}(\phi) = [-C, C]$ for some $C > 0$.

Now plug in this smooth version ℓ_h into (12) to obtain the following formula from (13):

$$\tilde{\boldsymbol{\beta}}_h^{/i} := \hat{\boldsymbol{\beta}}_h + \left[\textstyle\sum_{j\neq i} \boldsymbol{x}_j \boldsymbol{x}_j^\top \ddot{\ell}_h(\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}_h; y_j) + \nabla^2 R(\hat{\boldsymbol{\beta}}_h)\right]^{-1} \boldsymbol{x}_i \dot{\ell}_h(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}_h; y_i). \tag{15}$$

where $\hat{\boldsymbol{\beta}}_h$ is the minimizer on the full data from loss ℓ_h and R. $\tilde{\boldsymbol{\beta}}_h^{/i}$ is a good approximation to the leave-i-out estimator $\hat{\boldsymbol{\beta}}_h^{/i}$ based on smoothed loss ℓ_h. Setting $h \to 0$, we have $\ell_h(\mu, y)$ converge to $\ell(\mu, y)$ uniformly in the region of interest (see Appendix of [20] for the proof), implying that

$\lim_{h\to 0}\tilde{\boldsymbol{\beta}}_h^{/i}$ serves as a good estimator of $\lim_{h\to 0}\hat{\boldsymbol{\beta}}_h^{/i}$, which is close to the true leave-i-out $\hat{\boldsymbol{\beta}}^{/i}$. Equation (15) can be simplified in the limit $h \to 0$. We define the sets of indices V and S for the samples at singularities and smooth parts respectively:

$$V := \{j : \boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}} = v_t \text{ for some } t \in \{1, \dots, k\}\}, \qquad S := \{1, \dots, n\} \setminus V.$$

We characterize the limit of $\boldsymbol{x}_i^\top \tilde{\boldsymbol{\beta}}_h^{/i}$ below.

Theorem 3.1. *Under some mild conditions, as $h \to 0$, we have $\boldsymbol{x}_i^\top \tilde{\boldsymbol{\beta}}_h^{/i} \to \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}} + a_i g_{\ell,i}$ where*

$$\begin{cases} a_i = \frac{W_{ii}}{1 - W_{ii}\ddot{\ell}(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}; y_i)}, & g_{\ell,i} = \dot{\ell}(\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}; y_i) & \text{if } i \in S, \\ a_i = \frac{1}{[(\boldsymbol{X}_{V\cdot}\boldsymbol{Y}^{-1}\boldsymbol{X}_{V\cdot}^\top)^{-1}]_{ii}}, & g_{\ell,i} = \left[(\boldsymbol{X}_{V,\cdot}\boldsymbol{X}_{V,\cdot}^\top)^{-1}\boldsymbol{X}_{V,\cdot}[\nabla R(\hat{\boldsymbol{\beta}}) - \sum_{j\in S}\boldsymbol{x}_j \dot{\ell}(\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}}; y_j)]\right]_i & \text{if } i \in V, \end{cases}$$

with

$$\begin{aligned} \boldsymbol{Y} =& \nabla^2 R(\hat{\boldsymbol{\beta}}) + \boldsymbol{X}_{S\cdot}^\top \mathrm{diag}[\{\ddot{\ell}(\boldsymbol{x}_j^\top \hat{\boldsymbol{\beta}})\}_{j\in S}]\boldsymbol{X}_{S\cdot}, \\ W_{ii} =& \boldsymbol{x}_i^\top \boldsymbol{Y}^{-1}\boldsymbol{x}_i - \boldsymbol{x}_i^\top \boldsymbol{Y}^{-1}\boldsymbol{X}_{V,\cdot}^\top(\boldsymbol{X}_{V,\cdot}\boldsymbol{Y}^{-1}\boldsymbol{X}_{V,\cdot}^\top)^{-1}\boldsymbol{X}_{V,\cdot}\boldsymbol{Y}^{-1}\boldsymbol{x}_i. \end{aligned}$$

We can obtain the ALO estimate of prediction error by plugging $\boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}} + a_i g_{\ell,i}$ instead of $\boldsymbol{x}_i^\top \tilde{\boldsymbol{\beta}}^{/i}$ in (4). The conditions and proof of Theorem 3.1 can be found in Appendix of [20].

3.3 Equivalence Between Primal and Dual Methods

Although the primal and dual methods may be harder or easier to carry out depending on the specific problem at hand, one may wonder if they always obtain the same result. It turns out that there exists a unifying view for both methods, stated as below.

Suppose both ℓ and R are smooth. As both the primal and dual methods are based on a first-order approximation strategy, they are exact solutions to a surrogate quadratic leave-i-out problem. For the primal method, we have the following surrogate primal problem:

$$\min_{\boldsymbol{\beta}^{/i}} \sum_{j\neq i} \tilde{\ell}(\boldsymbol{x}_j^\top \boldsymbol{\beta}^{/i}; y_j) + \tilde{R}(\boldsymbol{\beta}^{/i}). \tag{16}$$

where $\tilde{\ell}$ and $\tilde{R}$ are the quadratic expansions of ℓ and R at $\hat{\boldsymbol{\beta}}$. The way we obtain $\tilde{\boldsymbol{\beta}}^{/i}$ in (13) indicates that the primal formula in (13) (14) are the exact leave-i-out solution of (16).

On the other hand, we may consider the surrogate dual problem, by replacing ℓ^* and R^* with their quadratic expansion at full data dual solution $\hat{\boldsymbol{\theta}}$ in the dual problem (6). It turns out that the surrogate dual problem is equivalent to the dual of the surrogate primal problem (16). In addition, the dual method described in Section 3 solves the surrogate dual problem. Therefore the primal and dual frameworks we laid out in Sections 3 lead to exactly the same ALO formulas. We again refer our reader to the full version of this paper [20] for more details about the proofs and the discussions of the nonsmooth situation.

4 Comparison of ALO and Infinitesimal Jackknife

Substantial effort has been devoted to the problem of parameter tuning in the past five decades. However, the incapability of the classical methods in addressing this problem in high-dimensional and big-data regimes has brought this problem back to the forefront of research [12, 6]. While due

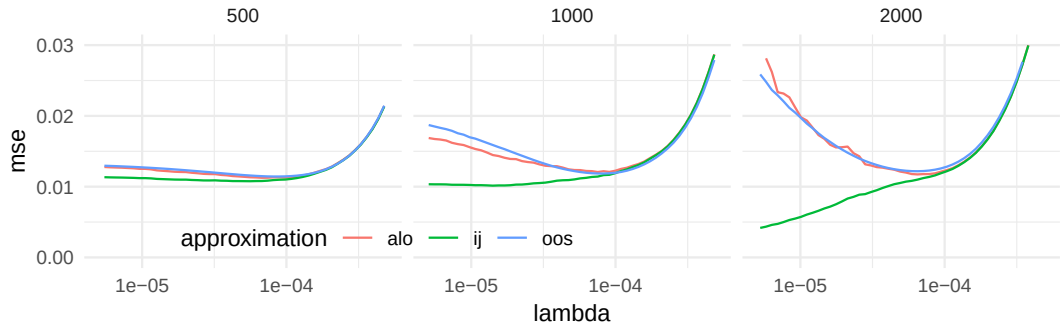

Figure 2: Risk estimates of different approximations for varying p ($n = 2000$, $k = 100$). Here, "alo" denotes the ALO estimator, "ij" the infinitesimal jacknife, and "oos" the true out-of-sample risk[2].

to space limitation we cannot mention all the recent proposals, we would like to compare our work with the most relevant one which is the infinitesimal jacknife (IJ) proposed in [6, 13]. We refer the interested reader to the full version of this paper [20] for more information on the other proposals.

IJ may be adapted to produce different approximate leave-one-out estimates of risk. For instance, in the case of LASSO, IJ yields $\frac{1}{n}\sum_{i=1}^{n}(\hat{r}_i^{\mathrm{IJ}})^2$ as an approximation of LOOCV[13], where

$$\hat{r}_i^{\mathrm{IJ}} = (1 + H_{ii})\hat{r}_i, \tag{17}$$

$\hat{r}_i = y_i - \boldsymbol{x}_i^\top \hat{\boldsymbol{\beta}}$, and $\boldsymbol{H} = \boldsymbol{X}_{\cdot,E}(\boldsymbol{X}_{\cdot,E}^\top \boldsymbol{X}_{\cdot,E})^{-1}\boldsymbol{X}_{\cdot,E}^\top$, with E denoting the equi-correlation set. In contrast, we have (see [20, Theorem 4.2]) $\mathrm{alo} = \frac{1}{n}\sum_{i=1}^{n}(\hat{r}_i^{\mathrm{ALO}})^2$, where $\hat{r}_i^{\mathrm{ALO}} = \frac{\hat{r}_i}{1-H_{ii}}$.

Note that when H_{ii} is small (close to zero), both estimates are essentially equivalent. However, in high dimensional models, we expect to have $H_{ii} = O(1)$ (in the case of LASSO, $\sum_i H_{ii} = |E|$), in which case the gap can be significant. In particular, when using the risk estimate for hyperparameter tuning, simulations show that IJ tends to select no regularization ($\lambda = 0$), whereas ALO can often produce a reasonable value (see Figure 2 and [13, Figure 5]).

5 Applications

5.1 Generalized LASSO

The generalized LASSO [17] is a generalization of the LASSO problem which captures many applications such as the fused LASSO [16], ℓ_1 trend filtering [9] and wavelet smoothing in a unified framework. The generalized LASSO problem solves the following penalized regression problem:

$$\min_{\boldsymbol{\beta}} \frac{1}{2}\sum_{j=1}^{n}(y_j - \boldsymbol{x}_j^\top \boldsymbol{\beta})^2 + \lambda\|\boldsymbol{D}\boldsymbol{\beta}\|_1. \tag{18}$$

where the regularizer is parameterized by a fixed matrix $\boldsymbol{D} \in \mathbb{R}^{m\times p}$ which captures the desired structure in the data. We note that the regularizer is a semi-norm. Hence we can formulate the dual problem as a projection. In fact, a dual formulation of (18) can be obtained as:

$$\min_{\boldsymbol{\theta},\boldsymbol{u}} \frac{1}{2}\|\boldsymbol{\theta} - \boldsymbol{y}\|_2^2 \quad \text{s.t. } \|\boldsymbol{u}\|_\infty \le \lambda \text{ and } \boldsymbol{X}^\top\boldsymbol{\theta} = \boldsymbol{D}^\top\boldsymbol{u}.$$

[2]The out-of-sample risk is the risk of the estimator under the data-generating distribution, which is known in the present case of a simulation.

The dual optimal solution satisfies $\hat{\boldsymbol{\theta}} = \Pi_{\Delta_X}(\boldsymbol{y})$, where Δ_X is the polytope given by:

$$\Delta_X = \{\boldsymbol{\theta} \in \mathbb{R}^n : \exists \boldsymbol{u}, \|u\|_\infty \leq \lambda \text{ and } \boldsymbol{X}^\top \boldsymbol{\theta} = \boldsymbol{D}^\top u\}.$$

The projection onto the polytope $C = \{\boldsymbol{D}^\top \boldsymbol{u} : \|\boldsymbol{u}\|_\infty \leq \lambda\}$ is given in [17] as locally being the projection onto the affine space orthogonal to the nullspace of $\boldsymbol{D}_{\cdot,-E}$, where $E = \{i : |\hat{u}_i| = \lambda\}$ and $-E = \{1, \ldots, p\} \setminus E$. Since $\Delta_X = [\boldsymbol{X}^\top]^{-1}C$ is the inverse image of C under the linear map given by $\boldsymbol{X}^\top$, the projection onto Δ_X is given locally by the projection onto the affine space normal to the space spanned by the columns of $[\boldsymbol{X}^\top]^+ \operatorname{null} \boldsymbol{D}_{\cdot,-E}$, provided $\boldsymbol{X}$ has full column rank. Here, $[\boldsymbol{X}^\top]^+$ denotes the Moore-Penrose pseudoinverse of $\boldsymbol{X}^\top$. To obtain a spanning set of this space, we consider $\boldsymbol{A} = \boldsymbol{X}\boldsymbol{B}$, where $\boldsymbol{B}$ is a set of vectors spanning the nullspace of $\boldsymbol{D}_{\cdot,-E}$. This allows us to compute $\boldsymbol{H} = \boldsymbol{A}\boldsymbol{A}^+$, the projection onto the normal space required to compute the ALO.

5.2 Kernel SVM

We present a derivation for the kernel formulation of SVM for classification. Let $\boldsymbol{K} \in \mathbb{R}^{n \times n}$ denote the kernel matrix (assumed positive-definite), and let $\boldsymbol{y} \in \{-1, 1\}^n$ denote the binary labels. The objective of the SVM (with no intercept) may be written as [19]:

$$\min_{\boldsymbol{\gamma}} \sum_{j=1}^{n} \big(1 - y_j h_j(\boldsymbol{\gamma}, \rho)\big)_+ + \frac{\lambda}{2} \boldsymbol{\gamma}^\top \boldsymbol{K} \boldsymbol{\gamma}, \quad \text{where } h_j(\boldsymbol{\gamma}) = \boldsymbol{K}_{\cdot,j}^\top \boldsymbol{\gamma}$$

As this objective is a combination of a smooth regularizer and a separable loss, we may apply Theorem 3.1 to obtain the approximation: $\tilde{y}_i^{/i} = \hat{y}_i + a_i g_i$, where a_i and g_i are defined as (here $\odot$ denotes component-wise multiplication):

$$a_i = \begin{cases} \lambda^{-1} \boldsymbol{K}_{\cdot,i}^\top \big[\boldsymbol{K}^{-1} - \boldsymbol{K}^{-1}\boldsymbol{K}_{\cdot,V}(\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}^{-1} \boldsymbol{K}_{\cdot,V})^{-1} \boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}^{-1}\big] \boldsymbol{K}_{\cdot,i} & \text{if } i \in S, \\ \big(\lambda \big[(\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}^{-1} \boldsymbol{K}_{\cdot,V})^{-1}\big]_{ii}\big)^{-1} & \text{if } i \in V, \end{cases}$$

$$\boldsymbol{g}_S = -\boldsymbol{y}_S \odot \mathbb{I}\{\boldsymbol{y}_S \boldsymbol{K}_{\cdot,S}^\top \hat{\boldsymbol{\gamma}} < 1\}, \quad \boldsymbol{g}_V = (\boldsymbol{K}_{\cdot,V}^\top \boldsymbol{K}_{\cdot,V})^{-1} \boldsymbol{K}_{\cdot,V}^\top \Big\{ \textstyle\sum_{j \in S : y_j \hat{y}_j < 1} y_j \boldsymbol{K}_{\cdot,j} - \lambda \boldsymbol{K}(\boldsymbol{y} \odot \hat{\boldsymbol{\gamma}}) \Big\}.$$

6 Numerical Experiments

We illustrate the performance of ALO through three experiments. The first two compare the ALO risk estimate with that of LOOCV. The third experiment compares the computational complexity of ALO with that of LOOCV. For the first experiment (Figure 3a), we run ALO and LOOCV for the two models studied in Section 5 (using fused LASSO as a special case of generalized LASSO) and compare their risk estimates under the settings $n > p$ and $n < p$ respectively. For the details of these experiments, the reader may refer to the full version of the paper [20].

For the second experiment (Figure 3b), we consider the risk estimates for LASSO from ALO and LOOCV under settings with model mis-specification, heavy-tail noise and correlated design. For all three cases, ALO approximates LOOCV well.

In general, we observe that the estimates given by ALO are close to LOOCV, although the performance may deteriorate for very small values of λ, as is clear in the fused-LASSO ($n < p$) example. These values of λ correspond to "dense" solutions, and are far from the optimal choice. Hence, such inaccuracies do not harm the parameter tuning algorithm.

Our last experiment compares the computational complexity of ALO with that of LOOCV. In Table 1, we provide the timing of LASSO for different values of n and p. The time required by ALO, which involves a single fit and a matrix inversion (in the construction of $\boldsymbol{H}$ matrix), is in all

Table 1: Timing (in *sec*) of one single fit, ALO and LOOCV.

(n, p)	(800, 200)	(800, 400)	(800, 1600)	(200, 800)	(400, 800)	(1600, 800)
single fit	0.035 ± 0.001	0.13 ± 0.01	0.60 ± 0.01	0.055 ± 0.002	0.19 ± 0.01	0.76 ± 0.02
ALO	0.060 ± 0.001	0.21 ± 0.01	0.89 ± 0.01	0.065 ± 0.001	0.24 ± 0.01	1.20 ± 0.01
LOOCV	27.52 ± 0.03	107.4 ± 0.5	479 ± 2	11.44 ± 0.049	74.7 ± 0.5	1249 ± 3

experiments no more than twice that of a single fit. We refer the reader to [20] for the details of all the above experiments.

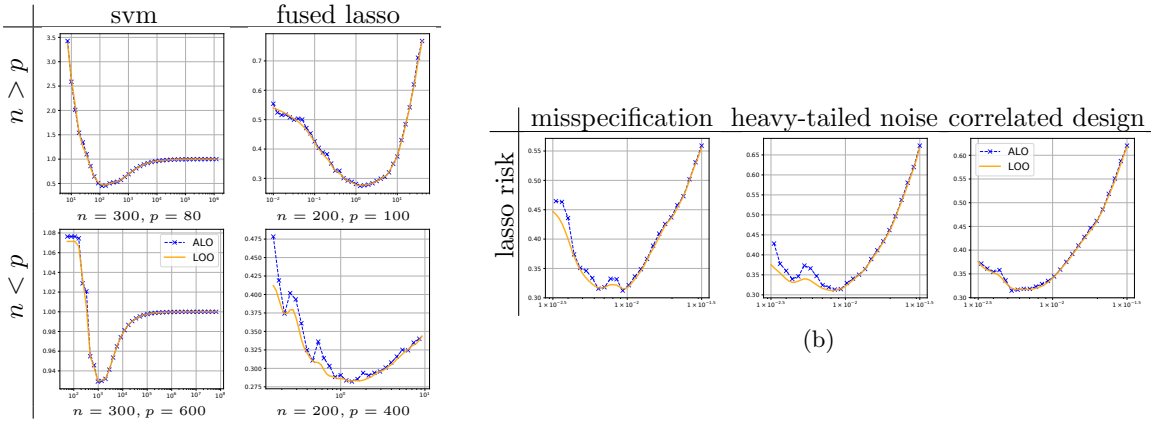

Figure 3: Risk estimates from ALO versus LOOCV. The x-axis is the value of λ on log-scale, the y-axis is the risk estimate. In part (a), the comparison is based on SVM and fused LASSO. In part (b), we consider the risk estimates of LASSO under model mis-specification, heavy-tailed noise and correlated design scenarios.

7 Discussion

ALO offers a highly efficient approach for parameter tuning and risk estimation for a large class of statistical machine learning models. We focus on nonsmooth models and propose two general frameworks for calculating ALO. One is from the primal perspective, the other from the dual.

By approximating LOOCV, ALO inherits desirable properties of LOOCV in high-dimensional settings where n and p are comparable. In particular, ALO can overcome the bias issues that k-fold cross validation displays in these settings.

Acknowledgements

We acknowledge computing resources from Columbia University's Shared Research Computing Facility project, which is supported by NIH Research Facility Improvement Grant 1G20RR030893-01, and associated funds from the New York State Empire State Development, Division of Science Technology and Innovation (NYSTAR) Contract C090171, both awarded April 15, 2010. We would also like to thank Linyun He, Wanchao Qin and Yuze Zhou for helpful discussion on computing ALO for kernel SVM.

References

[1] David M Allen. "The relationship between variable selection and data agumentation and a method for prediction". In: *Technometrics* 16.1 (1974), pp. 125–127.

[2] Dennis Amelunxen et al. "Living on the edge: Phase transitions in convex programs with random data". In: *Information and Inference: A Journal of the IMA* 3.3 (2014), pp. 224–294.

[3] Ahmad Beirami et al. "On Optimal Generalizability in Parametric Learning". In: *Advances in Neural Information Processing Systems*. 2017, pp. 3458–3468.

[4] Stephen Boyd and Lieven Vandenberghe. *Convex optimization*. Cambridge University Press, Cambridge, 2004, pp. xiv+716. ISBN: 0-521-83378-7.

[5] Gavin C Cawley and Nicola LC Talbot. "Efficient approximate leave-one-out cross-validation for kernel logistic regression". In: *Machine Learning* 71.2-3 (2008), pp. 243–264.

[6] Ryan Giordano et al. "A Swiss Army Infinitesimal Jackknife". In: *The 22nd International Conference on Artificial Intelligence and Statistics*. 2019, pp. 1139–1147.

[7] William W Hager. "Updating the inverse of a matrix". In: *SIAM review* 31.2 (1989), pp. 221–239.

[8] Peter J Huber. "Robust regression: asymptotics, conjectures and Monte Carlo". In: *The Annals of Statistics* (1973), pp. 799–821.

[9] Seung-Jean Kim et al. "l_1 trend filtering". In: *SIAM Rev.* 51.2 (2009), pp. 339–360.

[10] Rosa J Meijer and Jelle J Goeman. "Efficient approximate k-fold and leave-one-out cross-validation for ridge regression". In: *Biometrical Journal* 55.2 (2013), pp. 141–155.

[11] Manfred Opper and Ole Winther. "Gaussian processes and SVM: Mean field results and leave-one-out". In: (2000).

[12] Kamiar Rad and Arian Maleki. "A scalable estimate of the extra-sample prediction error via approximate leave-one-out". In: *arXiv preprint arXiv:1801.10243* (2018).

[13] William Stephenson and Tamara Broderick. "Sparse Approximate Cross-Validation for High-Dimensional GLMs". In: *arXiv preprint* (2019).

[14] Mervyn Stone. "Cross-validatory choice and assessment of statistical predictions". In: *Journal of the royal statistical society. Series B (Methodological)* (1974), pp. 111–147.

[15] Robert Tibshirani. "Regression shrinkage and selection via the lasso". In: *Journal of the Royal Statistical Society. Series B (Methodological)* (1996), pp. 267–288.

[16] Robert Tibshirani et al. "Sparsity and smoothness via the fused lasso". In: *J. R. Stat. Soc. Ser. B Stat. Methodol.* 67.1 (2005), pp. 91–108.

[17] Ryan J. Tibshirani and Jonathan Taylor. "The solution path of the generalized lasso". In: *Ann. Statist.* 39.3 (2011), pp. 1335–1371.

[18] Ryan J Tibshirani, Jonathan Taylor, et al. "Degrees of freedom in lasso problems". In: *The Annals of Statistics* 40.2 (2012), pp. 1198–1232.

[19] Grace Wahba, Yiing Yuh Lin, and Hansong Zhang. "Gacv for support vector machines". In: *Advances in Large Margin Classifiers*. 2000.

[20] Shuaiwen Wang et al. "Approximate Leave-One-Out for High-Dimensional Non-Differentiable Learning Problems". In: *arXiv preprint* (2018).