ROYAL STATISTICAL SOCIETY
DATA | EVIDENCE | DECISIONS

Journal of the Royal Statistical Society
Statistical Methodology
Series B

J. R. Statist. Soc. B (2020)
82, *Part* 4, *pp.* 965–996

A scalable estimate of the out-of-sample prediction error via approximate leave-one-out cross-validation

Kamiar Rahnama Rad

City University of New York, USA

and Arian Maleki

Columbia University, New York, USA

[Received May 2018. Final revision March 2020]

Summary. The paper considers the problem of out-of-sample risk estimation under the high dimensional settings where standard techniques such as K-fold cross-validation suffer from large biases. Motivated by the low bias of the leave-one-out cross-validation method, we propose a computationally efficient closed form approximate leave-one-out formula ALO for a large class of regularized estimators. Given the regularized estimate, calculating ALO requires a minor computational overhead. With minor assumptions about the data-generating process, we obtain a finite sample upper bound for the difference between leave-one-out cross-validation and approximate leave-one-out cross-validation, $|\text{LO} - \text{ALO}|$. Our theoretical analysis illustrates that $|\text{LO} - \text{ALO}| \longrightarrow 0$ with overwhelming probability, when $n, p \longrightarrow \infty$, where the dimension p of the feature vectors may be comparable with or even greater than the number of observations, n. Despite the high dimensionality of the problem, our theoretical results do not require any sparsity assumption on the vector of regression coefficients. Our extensive numerical experiments show that $|\text{LO} - \text{ALO}|$ decreases as n and p increase, revealing the excellent finite sample performance of approximate leave-one-out cross-validation. We further illustrate the usefulness of our proposed out-of-sample risk estimation method by an example of real recordings from spatially sensitive neurons (grid cells) in the medial entorhinal cortex of a rat.

Keywords: Cross-validation; Generalized linear models; High dimensional statistics; Out-of-sample risk estimation; Regularized estimation

1. Introduction

1.1. Main objectives

Consider a data set $\mathcal{D} = \{(y_1, \mathbf{x}_1), (y_2, \mathbf{x}_2), \ldots, (y_n, \mathbf{x}_n)\}$ where $\mathbf{x}_i \in R^p$ and $y_i \in R$. In many applications, we model these observations as independent and identically distributed draws from some joint distribution $q(y_i|\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*)\, p(\mathbf{x}_i)$ where the superscript 'T' denotes the transpose of a vector. To estimate the parameter $\boldsymbol{\beta}^*$ in such models, researchers often use the optimization problem

$$\hat{\boldsymbol{\beta}} \triangleq \arg\min_{\boldsymbol{\beta} \in R^p} \left\{ \sum_{i=1}^{n} l(y_i|\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \lambda r(\boldsymbol{\beta}) \right\}, \tag{1}$$

where l is called the loss function, and is typically set to $-\log\{q(y_i|\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})\}$ when q is known, and $r(\boldsymbol{\beta})$ is called the regularizer. In many applications, such as parameter tuning or model selection, one would like to estimate the *out-of-sample prediction error*, defined as

Address for correspondence: Kamiar Rahnama Rad, Paul H. Chook Department of Information Systems and Statistics, Zicklin School of Business, Baruch College, City University of New York, 55 Lexington Avenue at 24th Street, New York, NY 10010, USA.
E-mail: Kamiar.RahnamaRad@baruch.cuny.edu

 1369–7412/20/82965

$$\mathrm{Err}_{\mathrm{extra}} \triangleq E[\phi(y_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}}^{\mathrm{T}}\hat{\boldsymbol{\beta}})|\mathcal{D}], \tag{2}$$

where $(y_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}})$ is a new sample from the distribution $q(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}^*)p(\mathbf{x})$ independent of $\mathcal{D}$, and ϕ is a function that measures the closeness of y_{new} to $\mathbf{x}_{\mathrm{new}}^{\mathrm{T}}\hat{\boldsymbol{\beta}}$. A standard choice for ϕ is $-\log\{q(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\}$. However, in general we may use other functions as well. Since $\mathrm{Err}_{\mathrm{extra}}$ depends on the rarely known joint distribution of $(y_i, \mathbf{x}_i)$, a core problem in model assessment is to estimate it from data.

This paper considers a computationally efficient approach to the problem of estimating $\mathrm{Err}_{\mathrm{extra}}$ under the high dimensional setting, where both n and p are large, but n/p is a fixed number, possibly less than 1. This high dimensional setting has received much attention (El Karoui, 2018; El Karoui *et al.*, 2013; Bean *et al.*, 2013; Donoho and Montanari, 2016; Nevo and Ritov, 2016; Su *et al.*, 2017; Dobriban and Wager, 2018). But the problem of estimating $\mathrm{Err}_{\mathrm{extra}}$ has not been carefully studied in generality, and as a result the issues of the existing techniques and their remedies have not been explored. For instance, a popular technique in practice is K-fold cross-validation, where K is a small number, e.g. 3 or 5. Fig. 1 compares the performance of K-fold cross-validation for four values of K on a lasso linear regression problem. Fig. 1 implies that, in high dimensional settings, K-fold cross-validation suffers from a large bias, unless K is a large number. This bias is because of in high dimensional settings the fold that is removed in the training phase may have a major effect on the solution of problem (1). This claim can be easily seen for lasso linear regression with an independent and identically distributed data design matrix using phase transition diagrams (Donoho *et al.*, 2011). To summarize, as the number of folds increases, the bias of the estimates reduces at the expense of a higher computational complexity.

In this paper, we consider the most extreme form of cross-validation, namely leave-one-out cross-validation, which according to Fig. 1 is the least biased cross-validation-based estimate of the out-of-sample error. We shall use the fact that both n and p are large numbers to approximate leave-one-out cross-validation for both smooth and non-smooth regularizers. Our estimate, called approximate leave-one-out cross-validation, ALO requires solving optimization problem (1) once. Then, it uses $\hat{\boldsymbol{\beta}}$ to approximate leave-one-out cross-validation without solving the optimization problem again. In addition to obtaining $\hat{\boldsymbol{\beta}}$, approximate leave-one-out cross-validation requires a matrix inversion and two matrix–matrix multiplications. Despite these extra steps approximate leave-one-out cross-validation offers a significant computational saving compared with leave-one-out cross-validation. This point is illustrated in Fig. 2 by comparing the computational complexity of approximate leave-one-out cross-validation with that of leave-one-out cross-validation, LO, and a single fit as both n and p increase for various data shapes, i.e. $n > p$, $n = p$ and $n < p$. Details of this simulation are given in Section 5.2.4.

The main algorithmic and theoretical contributions of this paper are as follows. First, our computational complexity comparison between leave-one-out cross-validation and approximate leave-one-out cross-validation, confirmed by extensive numerical experiments, show that approximate leave-one-out cross-validation offers a major reduction in the computational complexity of estimating the out-of-sample risk. Moreover, with minor assumptions about the data-generating process, we obtain a finite sample upper bound $|\mathrm{LO} - \mathrm{ALO}|$, the difference between the leave-one-out and approximate leave-one-out cross-validation estimates, proving that under the high dimensional settings ALO presents a sensible approximation of LO for a large class of regularized estimation problems in the generalized linear family. Finally, we provide a readily usable R implementation of approximate leave-one-out cross-validation on line (see `https://github.com/Francis-Hsu/alocv`), and we illustrate the usefulness of our pro-

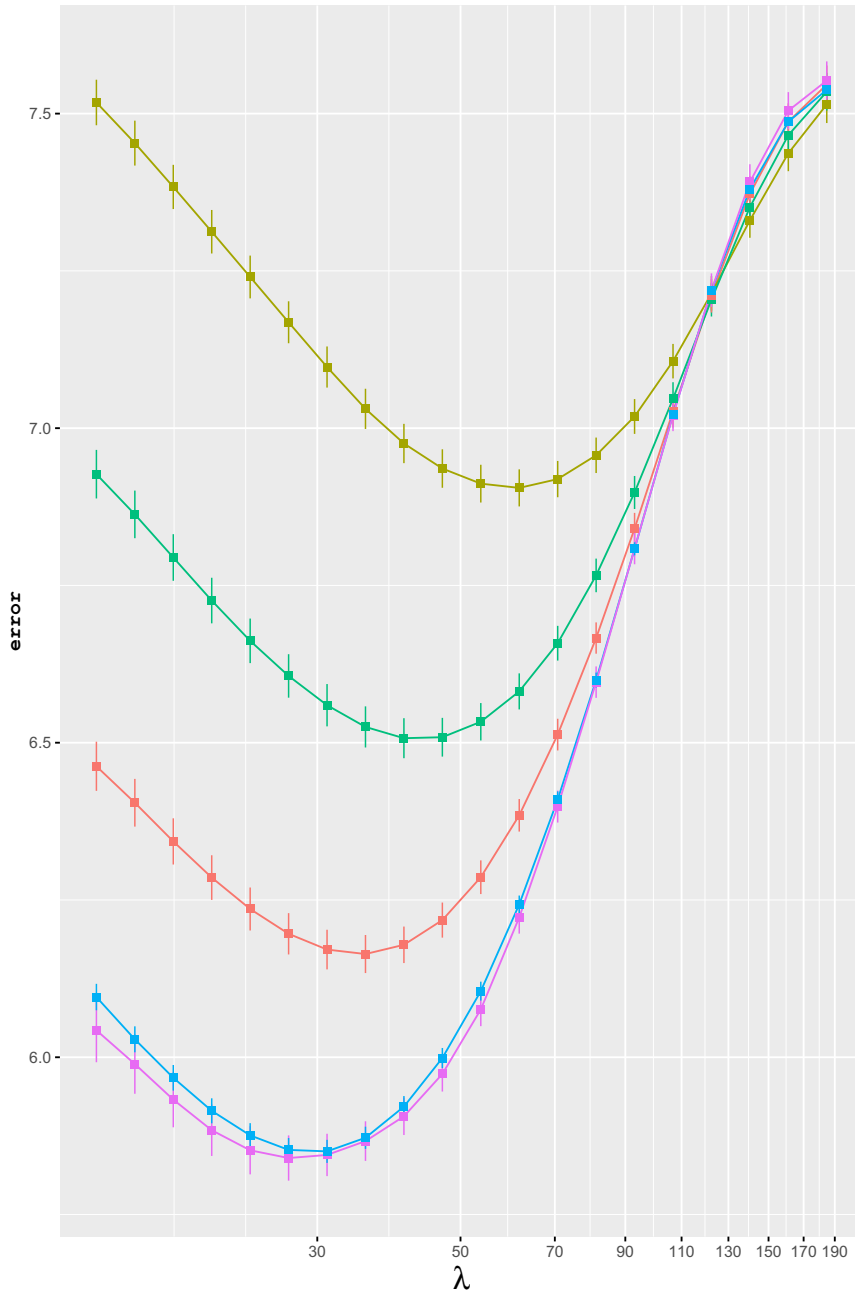

Fig. 1. Comparison of K-fold cross-validation (for $K = 3$ (■), 5 (■), 10 (■)) and leave-one-out cross-validation (■) with the true (oracle-based) out-of-sample error (■) for the lasso problem where $l(y|\mathbf{x}^\mathsf{T}\beta) = \frac{1}{2}(y - \mathbf{x}^\mathsf{T}\beta)^2$ and $r(\beta) = \|\beta\|_1$: in high dimensional settings the upward bias of K-fold cross-validation clearly decreases as the number of folds increases; the data are $\mathbf{y} \sim N(\mathbf{X}\beta^*, \sigma^2\mathbf{I})$ where $\mathbf{X} \in R^{p\times n}$; the number of non-zero elements of the true β^* is set to k and their values are set to $\frac{1}{3}$; the dimensions are $(p, n, k) = (1000, 250, 50)$ and $\sigma = 2$; the rows of $\mathbf{X}$ are independent $N(\mathbf{0}, \mathbf{I})$; the extra-sample test data are $y_{\text{new}} \sim N(\mathbf{x}_{\text{new}}^\mathsf{T}\beta^*, \sigma^2)$ where $\mathbf{x}_{\text{new}} \sim N(\mathbf{0}, \mathbf{I})$; the true (oracle-based) out-of-sample prediction error is $\text{Err}_{\text{extra}} = E[(y_{\text{new}} - \mathbf{x}_{\text{new}}^\mathsf{T}\hat{\beta})^2 |\mathbf{y}, \mathbf{X}] = \sigma^2 + \|\hat{\beta} - \beta^*\|_2^2$; all depicted quantities are averages based on 500 random independent samples, and error bars depict 1 standard error

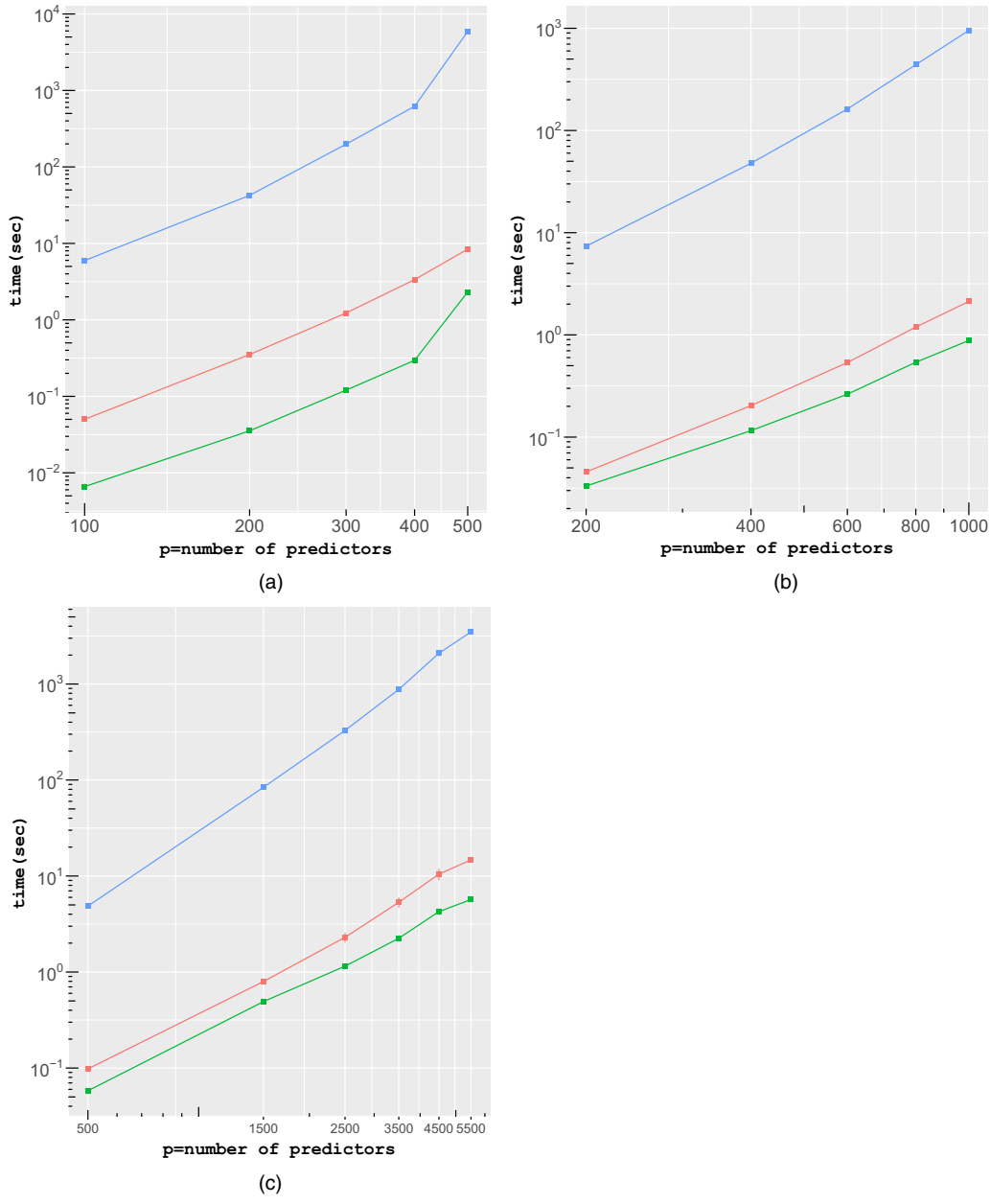

Fig. 2. Time to compute ALO (■) and LO (■), and the time to fit $\hat{\beta}$ (■) (the ALO-time includes computing $\hat{\beta}$; calculating LO takes orders of magnitude longer than ALO): (a) elastic net linear regression (Section 5.2.1) for $n/p = 5$; (b) lasso logistic regression (Section 5.2.2) for $n/p = 1$; (c) elastic net Poisson regression (Section 5.2.3) for $n/p = 1/10$

posed out-of-sample risk estimation in unexpected scenarios that fail to satisfy the assumptions of our theoretical framework. Specifically, we present a novel neuroscience example about the computationally efficient tuning of the spatial scale in estimating an inhomogeneous spatial point process.

1.2. Relevant work

The problem of estimating $\mathrm{Err}_{\mathrm{extra}}$ from $\mathcal{D}$ has been studied for (at least) the past 50 years. Methods such as cross-validation (Stone, 1974; Geisser, 1975), Allen's predicted residual error sum of squares statistic (Allen, 1974), generalized cross-validation (GCV) (Craven and Wahba, 1979; Golub *et al.*, 1979) and the bootstrap (Efron, 1983) have been proposed for this purpose. In the high dimensional setting, employing leave-one-out cross-validation or the bootstrap is computationally expensive and the less computationally complex approaches such as fivefold (or tenfold) cross-validation suffer from high bias as illustrated in Fig. 1.

As for the computationally efficient approaches, extensions of Allen's predicted residual error sum of squares (Allen, 1974) and GCV (Craven and Wahba, 1979; Golub *et al.*, 1979) to non-linear models and classifiers with a ridge penalty are well known: smoothing splines for generalized linear models (O'Sullivan *et al.*, 1986), spline estimation of generalized additive models (Burman, 1990), ridge estimators in logistic regression (le Cessie and van Houwelingen, 1992), smoothing splines with non-Gaussian data using various extensions of GCV (Gu, 1992, 2001; Xiang and Wahba, 1996), support vector machines (Opper and Winther, 2000), kernel logistic regression (Cawley and Talbot, 2008) and Cox's proportional hazard model with a ridge penalty (Meijer and Goeman, 2013). Moreover, leave-one-out approximations for posterior means of Bayesian models with Gaussian process priors by using the Laplace approximation and expectation propagation were introduced in Vehtari *et al.* (2016) and extended in Vehtari *et al.* (2017). Despite the existence of this vast literature, the performance of such approximations in high dimensional settings is unknown except for the straightforward linear ridge regression framework. Moreover, past heuristic approaches have considered only the ridge regularizer. The results of this paper include a much broader set of regularizers; examples include but are not limited to the lasso (Tibshirani, 1996), elastic net (Zou and Hastie, 2005) and bridge regression (Frank and Friedman, 1993), just to name a few.

More recently, a few papers have studied the problem of estimating $\mathrm{Err}_{\mathrm{extra}}$ under high dimensional settings (Mousavi *et al.*, 2018; Obuchi and Kabashima, 2016). The approximate message passing framework that was introduced in Maleki (2011) and Donoho *et al.* (2009) was used in Mousavi *et al.* (2018) to obtain an estimate of $\mathrm{Err}_{\mathrm{extra}}$ for lasso linear regression. In another related paper, Obuchi and Kabashima (2016) obtained similar results by using approximations that are popular in statistical physics. The results of Mousavi *et al.* (2018) and Obuchi and Kabashima (2016) are valid only for cases where the design matrix has independent and identically distributed entries and the empirical distribution of the regression coefficients converges weakly to a distribution with a bounded second moment. In this paper, our theoretical analysis includes correlated design matrices and regularized estimators beyond lasso linear regression.

In addition to these approaches, another contribution has been to study GCV and $\mathrm{Err}_{\mathrm{extra}}$ for restricted least squares estimators of submodels of the overall model without regularization (Breiman and Freedman, 1983; Leeb, 2008, 2009). In Leeb (2008) it was shown that a variant of GCV converges to $\mathrm{Err}_{\mathrm{extra}}$ uniformly over a collection of candidate models provided that there are not too many candidate models, ruling out complete subset selection. Moreover, since restricted least squares estimators were studied, the conclusions exclude the regularized problems that are considered in this paper.

Finally, it is worth mentioning that, in another line of work, strategies have been proposed to obtain unbiased estimates of the in-sample error. In contrast with the out-of-sample error, the in-sample error is about the prediction of new responses for the same explanatory variables as in the training data. The literature of in-sample error estimation is too vast to be reviewed here. Mallows's C_{p} (Mallows, 1973), Akaike's information criterion (Akaike, 1974; Hurvich and Tsai,

1989), Stein's unbiased risk estimate (Stein, 1981; Zou *et al.*, 2007; Tibshirani and Taylor, 2012) and Efron's covariance penalty (Efron, 1986) are seminal examples of in-sample error estimators. When n is much larger than p, the in-sample prediction error is expected to be close to the out-of-sample prediction error. The problem is that in high dimensional settings, where n is of the same order as (or even smaller than) p, the in-sample and out-of-sample errors are different.

The rest of the paper is organized as follows. After introducing the notation, we first present the approximate leave-one-out formula for twice differentiable regularizers in Section 2.1. In Section 2.2. we show how approximate leave-one-out cross-validation can be extended to non-smooth regularizers such as the lasso by using theorem 1 and theorem 2. In Section 3, we compare the computational complexity and memory requirements of approximate leave-one-out cross-validation and leave-one-out cross-validation. In Section 4, we present theorem 3, illustrating with minor assumptions about the data-generating process that $|\mathrm{LO} - \mathrm{ALO}| \to 0$ with overwhelming probability, when $n, p \to \infty$, where p may be comparable with or even greater than n. The numerical examples in Section 5 study the statistical accuracy and computational efficiency of the approximate leave-one-out approach. To illustrate the accuracy and computational efficiency of approximate leave-one-out cross-validation we apply it to synthetic and real data in Section 5. We generate synthetic data, and compare approximate leave-one-out cross-validation and leave-one-out cross-validation with elastic net linear regression in Section 5.2.1, lasso logistic regression in Section 5.2.2 and elastic net Poisson regression in Section 5.2.3. For real data we apply the lasso, elastic net and ridge logistic regression to sonar returns from two undersea targets in Section 5.3.1, and we apply lasso Poisson regression to real recordings from spatially sensitive neurons (grid cells) in Section 5.3.2. Our synthetic and real data examples cover various data shapes, i.e. $n > p$, $n = p$ and $n < p$. In Section 6 we discuss directions for future work. Technical proofs are collected in section A of the on-line appendix.

1.3. Notation

We first review the notation that will be used in the rest of the paper. Let $\mathbf{x}_i^{\mathrm{T}} \in R^{1\times p}$ stand for the ith row of $\mathbf{X} \in R^{n\times p}$. $\mathbf{y}_{/i} \in R^{(n-1)\times 1}$ and $\mathbf{X}_{/i} \in R^{(n-1)\times p}$ stand for $\mathbf{y}$ and $\mathbf{X}$, excluding the ith entry y_i and the ith row $\mathbf{x}_i^{\mathrm{T}}$ respectively. The vector $\mathbf{a} \odot \mathbf{b}$ stands for the entrywise product of two vectors $\mathbf{a}$ and $\mathbf{b}$. For two vectors $\mathbf{a}$ and $\mathbf{b}$, we use $\mathbf{a} < \mathbf{b}$ to indicate elementwise inequalities. Moreover, $|\mathbf{a}|$ stands for the vector that is obtained by applying the elementwise absolute value to every element of $\mathbf{a}$. For a set $S \subset \{1, 2, 3, \ldots, p\}$, let $\mathbf{X}_S$ stand for the submatrix of $\mathbf{X}$ restricted to *columns* indexed by S. Likewise, we let $\mathbf{x}_{i,S} \in R^{|S|\times 1}$ stand for the subvector of $\mathbf{x}_i$ restricted to the entries that are indexed by S. For a vector $\mathbf{a}$, depending on which notation is easier to read, we may use $[\mathbf{a}]_i$ or a_i to denote the ith entry of $\mathbf{a}$. The diagonal matrix with elements of the vector $\mathbf{a}$ is referred to as $\mathrm{diag}(\mathbf{a})$. Moreover, define

$$\phi(y, z) \triangleq \frac{\partial \phi(y, z)}{\partial z},$$

$$\dot{l}_i(\boldsymbol{\beta}) \triangleq \frac{\partial l(y_i|z)}{\partial z}|_{z=\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}},$$

$$\ddot{l}_i(\boldsymbol{\beta}) \triangleq \frac{\partial^2 l(y_i|z)}{\partial z^2}|_{z=\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}},$$

$$\dot{\mathbf{l}}_{/i}(\cdot) \triangleq (\dot{l}_1(\cdot), \ldots, \dot{l}_{i-1}(\cdot), \dot{l}_{i+1}(\cdot), \ldots, \dot{l}_n(\cdot))^{\mathrm{T}},$$

$$\ddot{\mathbf{l}}_{/i}(\cdot) \triangleq (\ddot{l}_1(\cdot), \ldots, \ddot{l}_{i-1}(\cdot), \ddot{l}_{i+1}(\cdot), \ldots, \ddot{l}_n(\cdot))^{\mathrm{T}}.$$

The notation $\mathrm{PolyLog}(n)$ denotes a polynomial of $\log(n)$ with a finite degree. Finally, let $\sigma_{\max}(\mathbf{A})$ and $\sigma_{\min}(\mathbf{A})$ stand for the largest and smallest singular values of $\mathbf{A}$ respectively.

2. Approximate leave-one-out cross-validation

2.1. Twice differentiable losses and regularizers

The leave-one-out cross-validation estimate is defined through the following formula:

$$\mathrm{LO} \triangleq \frac{1}{n}\sum_{i=1}^{n}\phi(y_i, \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i}), \tag{3}$$

where

$$\hat{\boldsymbol{\beta}}_{/i} \triangleq \underset{\boldsymbol{\beta}\in R^p}{\arg\min}\Big\{\sum_{j\neq i} l(y_j|\mathbf{x}_j^{\mathrm{T}}\boldsymbol{\beta}) + \lambda r(\boldsymbol{\beta})\Big\}, \tag{4}$$

is the leave-i-out estimate. If done naively, the calculation of LO asks for the optimization problem (4) to be solved n times, which is a computationally demanding task when p and n are large. To resolve this issue, we use the following simple strategy: instead of solving problem (4) accurately, we use one step of the Newton method for solving problem (4) with initialization $\hat{\boldsymbol{\beta}}$. Note that this step requires both l and r to be twice differentiable. We shall explain how this limitation can be lifted in the next section. The Newton step leads to the following simple approximation of $\hat{\boldsymbol{\beta}}_{/i}$:

$$\tilde{\boldsymbol{\beta}}_{/i} = \hat{\boldsymbol{\beta}} + \Big[\sum_{j\neq i}\mathbf{x}_j\mathbf{x}_j^{\mathrm{T}}\ddot{l}(y_j|\mathbf{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}) + \lambda\,\mathrm{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\}\Big]^{-1}\mathbf{x}_i\dot{l}(y_i|\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}),$$

where $\hat{\boldsymbol{\beta}}$ is defined in equation (1). (In the rest of the paper for notational simplicity of our theoretical results we have assumed that $r(\boldsymbol{\beta}) = \Sigma_{i=1}^{p} r(\beta_i)$. However, the extension to non-separable regularizers is straightforward.) Note that $\Sigma_{j\neq i}\mathbf{x}_j\mathbf{x}_j^{\mathrm{T}}\ddot{l}(y_j|\mathbf{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}) + \lambda\,\mathrm{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\}$ is still dependent on the observation that is removed. Hence, the process of computing the inverse (or solving a linear equation) must be repeated n times. Standard methods for calculating inverses (or solving linear equations) require cubic time and quadratic space (see appendix C.3 in Boyd and Vandenberghe (2004)), rendering them impractical for high dimensional applications when repeated n times. (A natural idea for reducing the computational burden involves exploiting structures (such as sparsity and bandedness) of the matrices involved. However, in this paper we do not make any assumption regarding the structure of $\mathbf{X}$.) We use the Woodbury lemma to reduce the computational cost:

$$\Big[\sum_{j\neq i}\mathbf{x}_j\mathbf{x}_j^{\mathrm{T}}\ddot{l}(y_j|\mathbf{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}) + \lambda\,\mathrm{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\}\Big]^{-1} = \mathbf{J}^{-1} + \frac{\mathbf{J}^{-1}\mathbf{x}_i\ddot{l}(y_i|\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}})\mathbf{x}_i^{\mathrm{T}}\mathbf{J}^{-1}}{1-\mathbf{x}_i^{\mathrm{T}}\mathbf{J}^{-1}\mathbf{x}_i\ddot{l}(y_i|\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}})}, \tag{5}$$

where $\mathbf{J} = \Sigma_{j=1}^{n}\mathbf{x}_j\mathbf{x}_j^{\mathrm{T}}\ddot{l}(y_j|\mathbf{x}_j^{\mathrm{T}}\hat{\boldsymbol{\beta}}) + \lambda\,\mathrm{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\}$. Following this approach we define the approximate leave-one-out cross-validation estimate ALO as

$$\mathrm{ALO} \triangleq \frac{1}{n}\sum_{i=1}^{n}\phi(y_i, \mathbf{x}_i^{\mathrm{T}}\tilde{\boldsymbol{\beta}}_{/i}) = \frac{1}{n}\sum_{i=1}^{n}\phi\Big\{y_i, \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})}\frac{H_{ii}}{1-H_{ii}}\Big\}, \tag{6}$$

where

$$\mathbf{H} \triangleq \mathbf{X}[\lambda\,\mathrm{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\} + \mathbf{X}^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}]^{-1}\mathbf{X}^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}. \tag{7}$$

Algorithm 1 (Table 1) summarizes how one should obtain an ALO-estimate of $\mathrm{Err}_{\mathrm{extra}}$. We shall show that under the high dimensional settings one Newton step is sufficient for obtaining a good approximation of $\hat{\boldsymbol{\beta}}_{/i}$, and the difference $|\mathrm{ALO} - \mathrm{LO}|$ is small when either n or both

Table 1. Algorithm 1: risk estimation with ALO for twice-differentiable losses and regularizers

Input: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$
Output: $\mathrm{Err}_{\mathrm{extra}}$-estimate
Step 1: calculate

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in R^p} \left\{ \sum_{i=1}^{n} l(y_i | \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda r(\boldsymbol{\beta}) \right\}$$

Step 2: obtain

$$\mathbf{H} = \mathbf{X}[\lambda \operatorname{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\} + \mathbf{X}^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}]^{-1} \mathbf{X}^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}$$

Step 3: the estimate of $\mathrm{Err}_{\mathrm{extra}}$ is given by

$$\frac{1}{n} \sum_{i=1}^{n} \phi \left\{ y_i, \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}}{1 - H_{ii}} \right\}$$

n and p are large. However, before that we resolve the differentiability issue of the approach that we discussed above.

2.2. Non-smooth regularizers

The Newton step, which was used in the derivation of ALO, requires the twice differentiability of the loss function and regularizer. However, in many modern applications non-smooth regularizers, such as the lasso, are preferable. In this section, we explain how ALO can be used for non-smooth regularizers. We start with the l_1-regularizer and then extend it to the other bridge estimators. A similar approach can be used for other non-smooth regularizers. Consider

$$\hat{\boldsymbol{\beta}} \triangleq \arg\min_{\boldsymbol{\beta} \in R^p} \left\{ \sum_{i=1}^{n} l(y_i | \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda \|\boldsymbol{\beta}\|_1 \right\}. \tag{8}$$

Let $\hat{\mathbf{g}}$ be a subgradient of $\|\boldsymbol{\beta}\|_1$ at $\hat{\boldsymbol{\beta}}$, denoted by $\hat{\mathbf{g}} \in \partial \|\hat{\boldsymbol{\beta}}\|_1$. Then, the pair $(\hat{\boldsymbol{\beta}}, \hat{\mathbf{g}})$ must satisfy the zero-subgradient condition

$$\sum_{i=1}^{n} \mathbf{x}_i \dot{l}(y_i | \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}}) + \lambda \hat{\mathbf{g}} = 0.$$

As a starting point we use a smooth approximation of the function $\|\boldsymbol{\beta}\|_1$ in our ALO-formula. For instance, we can use the following approximation that was introduced in Schmidt *et al.* (2007):

$$r^{\alpha}(\boldsymbol{\beta}) = \sum_{i=1}^{p} \frac{1}{\alpha} [\log\{1 + \exp(\alpha \beta_i)\} + \log\{1 + \exp(-\alpha \beta_i)\}].$$

Since $\lim_{\alpha \to \infty} r^{\alpha}(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$, we can use

$$\hat{\boldsymbol{\beta}}^{\alpha} \triangleq \arg\min_{\boldsymbol{\beta} \in R^p} \left\{ \sum_{i=1}^{n} l(y_i | \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda \sum_{i=1}^{p} r^{\alpha}(\beta_i) \right\}, \tag{9}$$

to obtain the following formula for ALO:

$$\mathrm{ALO}^{\alpha} \triangleq \frac{1}{n} \sum_{i=1}^{n} \phi \left\{ y_i, \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}}^{\alpha} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}}^{\alpha})}{\ddot{l}_i(\hat{\boldsymbol{\beta}}^{\alpha})} \frac{H_{ii}^{\alpha}}{1 - H_{ii}^{\alpha}} \right\} \tag{10}$$

where $\mathbf{H}^{\alpha} \triangleq \mathbf{X}[\lambda \operatorname{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}}^{\alpha})\} + \mathbf{X}^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}}^{\alpha})\}\mathbf{X}]^{-1} \mathbf{X}^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}}^{\alpha})\}$. Note that $\|\hat{\boldsymbol{\beta}}^{\alpha} - \hat{\boldsymbol{\beta}}\|_2 \to 0$ as

$\alpha \to \infty$, according to lemma 15 in the on-line appendix section A.2. Therefore, we take the $\alpha \to \infty$ limit in expression (10), yielding a simplification of ALO^{α} in this limit. To prove this claim, we denote the active set of $\hat{\boldsymbol{\beta}}$ with S, and we suppose the following assumptions.

Assumption 1. $\hat{\boldsymbol{\beta}}$ is the unique global minimizer of problem (1).

Assumption 2. $\hat{\boldsymbol{\beta}}^{\alpha}$ is the unique global minimizer of problem (9) for every value of α.

Assumption 3. $\ddot{l}(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$ is a continuous function of $\boldsymbol{\beta}$.

Assumption 4. The strict dual feasibility condition $\|\hat{\mathbf{g}}_{S^{\mathrm{c}}}\|_{\infty} < 1$ holds.

Theorem 1. If assumptions 1–4 hold, then

$$\lim_{\alpha \to \infty} \mathrm{ALO}^{\alpha} = \frac{1}{n} \sum_{i=1}^{n} \phi\left\{ y_i, \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}}{1 - H_{ii}} \right\}, \tag{11}$$

where $\mathbf{H} = \mathbf{X}_S[\mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_S]^{-1}\mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}$.

The proof of theorem 1 is presented in the on-line appendix section A.2. For the rest of the paper, the right-hand side of result (11) is the ALO-formula that we use as an approximation of LO for lasso problems. In Section 5.2, we show that the formula that we obtain in theorem 1 offers an accurate estimate of the out-of-sample prediction error. For instance, in the standard lasso problem, where $l(u, v) = (u - v)^2/2$ and $r(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$, theorem 1 gives the following estimate of the out-of-sample prediction error:

$$\lim_{\alpha \to \infty} \mathrm{ALO}^{\alpha} = \frac{1}{n} \sum_{i=1}^{n} \frac{(y_i - \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}})^2}{(1 - H_{ii})^2}, \tag{12}$$

where $\mathbf{H} = \mathbf{X}_S(\mathbf{X}_S^{\mathrm{T}}\mathbf{X}_S)^{-1}\mathbf{X}_S^{\mathrm{T}}$. Fig. 3 compares this estimate with the oracle estimate of the out-of-sample prediction error on a lasso example. More extensive simulations are reported in Section 5.

Assumptions 1–3 hold for most of the practical problems. For instance, to study the conditions under which assumption 1 holds refer to Tibshirani (2013). Moreover, for $l(u, v) = (u - v)^2/2$, assumption 1 is a consequence of assumption 4 (Wainwright, 2009). Assumption 4 also holds in many cases with probability 1 with respect to the randomness of the data set (Wainwright, 2009; Tibshirani and Taylor, 2012). Even if this assumption is violated in a specific problem (note that checking this assumption is straightforward), we can use the following theorem to evaluate the accuracy of the ALO-formula in theorem 1.

Theorem 2. Let S and T denote the active set of $\hat{\boldsymbol{\beta}}$, and the set of zero coefficients at which the subgradient vector is equal to 1 or -1. Then,

$$\mathbf{x}_{i,S}^{\mathrm{T}}[\mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_S]^{-1}\mathbf{x}_{i,S}\ddot{l}_i(\hat{\boldsymbol{\beta}}) < \lim \inf_{\alpha \to \infty} H_{ii}^{\alpha},$$

$$\lim \sup_{\alpha \to \infty} H_{ii}^{\alpha} < \mathbf{x}_{i,S\cup T}^{\mathrm{T}}[\mathbf{X}_{S\cup T}^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_{S\cup T}]^{-1}\mathbf{x}_{i,S\cup T}\ddot{l}_i(\hat{\boldsymbol{\beta}}).$$

Theorem 2 is proved in the on-line appendix section A.3. A simple implication of theorem 2 is that

$$\lim \sup_{\alpha \to \infty} \mathrm{ALO}^{\alpha} \leqslant \frac{1}{n} \sum_{i=1}^{n} \phi\left\{ y_i, \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}^{h}}{1 - H_{ii}^{h}} \right\}, \tag{13}$$

and

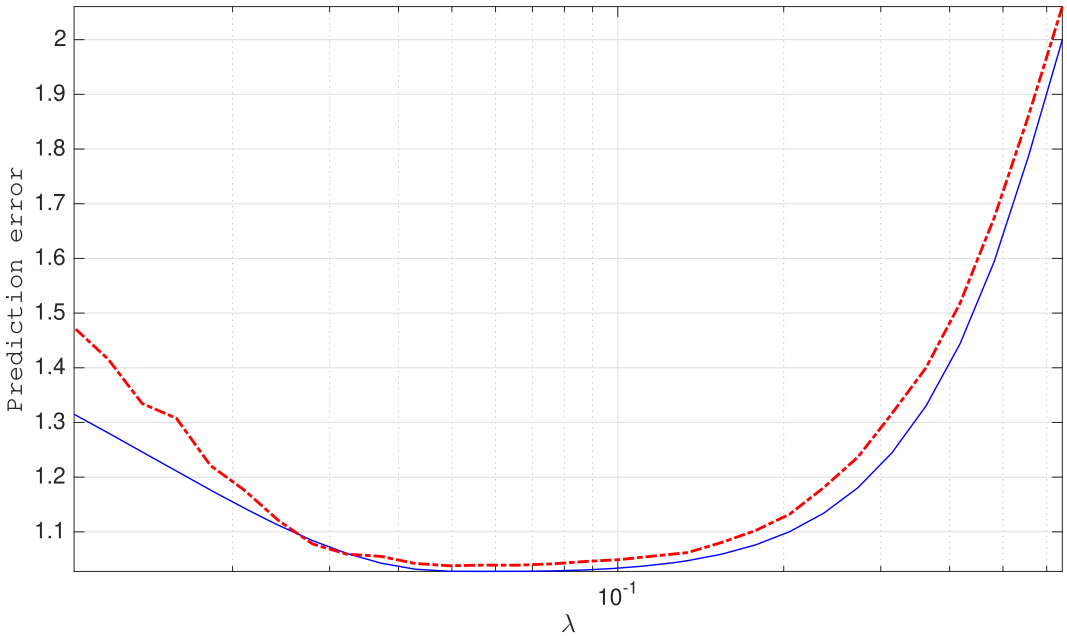

Fig. 3. Out-of-sample prediction error *versus* ALO (———, $\mathrm{Err}_{\mathrm{extra},\lambda}$; - - - - -, ALO_λ): the data are $\mathbf{y} \sim N(\mathbf{X}\beta^*, \sigma^2\mathbf{I})$ where $\sigma^2 = 1$ and $\mathbf{X} \in R^{p\times n}$ with $p = 10000$ and $n = 2000$; the number of non-zero elements of the true β^* is set to $k = 400$ and their values are set to 1; the rows $\mathbf{x}_i^{\mathrm{T}}$ of the predictor matrix are generated randomly as $N(0, \Sigma)$ with correlation structure $\mathrm{corr}(X_{ij}, X_{ij'}) = 0.3$ for all $i = 1, \ldots, n$ and $j, j' = 1, \ldots, p$; the covariance matrix Σ is scaled such that the signal variance $\mathrm{var}(\mathbf{x}^{\mathrm{T}}\beta^*) = 1$; the out-of-sample test data are $y_{\mathrm{new}} \sim N(\mathbf{x}_{\mathrm{new}}^{\mathrm{T}}\beta^*, \sigma^2)$ where $\mathbf{x}_{\mathrm{new}} \sim N(0, \Sigma)$; the out-of-sample error is calculated as $E_{(y_{\mathrm{new}}, \mathbf{x}_{\mathrm{new}})}[(y_{\mathrm{new}} - \mathbf{x}_{\mathrm{new}}^{\mathrm{T}}\hat{\beta})^2 | \mathbf{y}, \mathbf{X}] = \sigma^2 + \|\Sigma^{1/2}(\hat{\beta} - \beta^*)\|_2^2$ and ALO is calculated by using equation (12)

$$\lim \inf_{\alpha\to\infty} \mathrm{ALO}^\alpha \geqslant \frac{1}{n}\sum_{i=1}^{n} \phi\left\{ y_i, \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}^l}{1 - H_{ii}^l} \right\}, \tag{14}$$

where

$$\mathbf{H}^l = \mathbf{X}_S[\mathbf{X}_S^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_S]^{-1}\mathbf{X}_S^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\},$$

$$\mathbf{H}^h = \mathbf{X}_{S\cup T}[\mathbf{X}_{S\cup T}^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_{S\cup T}]^{-1}\mathbf{X}_{S\cup T}^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}. \tag{15}$$

By comparing inequalities (13) and (14) we can evaluate the error in our simple formula of the risk, presented in theorem 1. The approach that we proposed above can be extended to other non-differentiable regularizers as well. Below we consider two other popular classes of estimators:

(a) bridge and
(b) elastic net,

and show how we can derive ALO-formulae for each estimator.

2.2.1. *Bridge estimators*

Consider the class of bridge estimators

$$\hat{\boldsymbol{\beta}} \triangleq \arg\min_{\boldsymbol{\beta}\in R^p} \left\{ \sum_{i=1}^{n} l(y_i | \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) + \lambda\|\boldsymbol{\beta}\|_q^q \right\}, \tag{16}$$

Table 2. Algorithm 2 risk estimation with ALO for the elastic net regularizer

Input: $(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)$
Output: $\mathrm{Err}_{\mathrm{extra}}$-estimate
Step 1: calculate

$$\hat{\boldsymbol{\beta}} = \arg\min_{\boldsymbol{\beta} \in R^p} \left\{ \sum_{i=1}^{n} l(y_i | \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda_1 \|\boldsymbol{\beta}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_1 \right\}$$

Step 2: calculate $S = \{i : \hat{\beta}_i \neq 0\}$
Step 3: obtain $\mathbf{H} = \mathbf{X}_S [\mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\} \mathbf{X}_S + 2\lambda_1 \mathbf{I}]^{-1} \mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}$, where $\mathbf{X}_S$ includes only the columns of $\mathbf{X}$ that are in S
Step 4: the estimate of $\mathrm{Err}_{\mathrm{extra}}$ is given by

$$\frac{1}{n} \sum_{i=1}^{n} \phi \left\{ y_i, \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}}{1 - H_{ii}} \right\}$$

where q is a number between $(1, 2)$. Note that these regularizers are only one-time differentiable at zero. Hence, the Newton method that was introduced in Section 2.1 is not directly applicable. One can argue intuitively that, since the regularizer is differentiable at zero, none of the regression coefficients will be 0. Hence, the regularizer is locally twice differentiable and formula (6) works well. Although this argument is often correct, we can again use the idea that was introduced above for the lasso to obtain the following ALO-formula that can be used even when an estimate of 0 is observed:

$$\frac{1}{n} \sum_{i=1}^{n} \phi \left\{ y_i, \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}}{1 - H_{ii}} \right\}, \tag{17}$$

where, if we define $S =^{\triangle} \{i : \beta_i \neq 0\}$ and for $u \neq 0$, $\ddot{r}^q(u) =^{\triangle} q(q-1)|u|^{q-2}$, then

$$\mathbf{H} = \mathbf{X}_S [\mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\} \mathbf{X}_S + \lambda \, \mathrm{diag}\{\ddot{\mathbf{r}}_S^q(\hat{\boldsymbol{\beta}})\}]^{-1} \mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}. \tag{18}$$

This formula is derived in the on-line appendix section A.4.

2.2.2. Elastic net

Finally, we consider the elastic net estimator

$$\hat{\boldsymbol{\beta}} \triangleq \arg\min_{\boldsymbol{\beta} \in R^p} \left\{ \sum_{i=1}^{n} l(y_i | \mathbf{x}_i^{\mathrm{T}} \boldsymbol{\beta}) + \lambda_1 \|\boldsymbol{\beta}\|_2^2 + \lambda_2 \|\boldsymbol{\beta}\|_1 \right\}. \tag{19}$$

Again by smoothing the l_1-regularizer (similarly to what we did for the lasso) we obtain the following ALO-formula for the out-of-sample predictor error:

$$\frac{1}{n} \sum_{i=1}^{n} \phi \left\{ y_i, \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}}{1 - H_{ii}} \right\},$$

where $S = \{i : \hat{\beta}_i \neq 0\}$, and

$$\mathbf{H} = \mathbf{X}_S [\mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\} \mathbf{X}_S + 2\lambda_1 \mathbf{I}]^{-1} \mathbf{X}_S^{\mathrm{T}} \mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}. \tag{20}$$

We do not derive this formula, since it follows exactly the same lines as those of the lasso and the bridge estimator. Algorithm 2 (Table 2) summarizes all the calculations that are required for the calculation of ALO for the elastic net.

3. Computational complexity and memory requirements of approximate leave-one-out cross-validation

Counting the number of floating point operations that algorithms require is a standard approach for comparing their computational complexities. In this section, we calculate and compare the number of operations that are required by ALO and LO. We first start with algorithm 1 and then discuss algorithm 2.

3.1. Algorithm 1

Before we start the calculations, we warn the reader that in many cases the specific structure of the loss and/or the regularizer enables more efficient implementation of the formulae. However, here we consider the worst-case scenario. Furthermore, the calculations below are concerned with the implementation of ALO and LO on a single computer, and we have not explored their parallel or distributed implementations.

The first step of algorithm 1 requires solving an optimization problem. Several methods exist for solving this optimization problem. Here, we discuss the interior point method and the accelerated gradient descent algorithm. Suppose that our goal is to reach accuracy ϵ. Then, the interior point method requires $O\{\log(1/\epsilon)\}$ iterations to reach this accuracy, whereas accelerated gradient descent requires $O(1/\sqrt{\epsilon})$ iterations (Nesterov, 2013). Furthermore, each iteration of the accelerated gradient descent requires $O(np)$ operations, whereas each iteration of the interior point method requires $O(p^3)$ operations.

Regarding the memory usage of these two algorithms, in the accelerated gradient descent algorithm the memory is mainly used for storing matrix $\mathbf{X}$. Hence, the amount of memory that is required by this algorithm is $O(np)$. In contrast, the interior point method uses $O(p^3)$ of memory.

The second step of algorithm 1 is to calculate the matrix $\mathbf{H}$. This requires inverting the matrix $[\lambda \operatorname{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\} + \mathbf{X}^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}]^{-1}$. In general, this inversion requires $O(p^3)$ operations (e.g. by using Cholesky factorization). However, if n is much smaller than p, then one can use a better trick for performing the matrix inversion; suppose that both l and r are strongly convex at $\hat{\boldsymbol{\beta}}$ and define $\boldsymbol{\Gamma} =^{\triangle} [\operatorname{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}]^{1/2}$, and $\boldsymbol{\Lambda} =^{\triangle} \lambda \operatorname{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\}$. Then, from the matrix inversion lemma we have

$$\mathbf{X}(\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma}^2\mathbf{X} + \boldsymbol{\Lambda})^{-1}\mathbf{X}^{\mathrm{T}} = \mathbf{X}\boldsymbol{\Lambda}^{-1}\mathbf{X}^{\mathrm{T}} - \mathbf{X}\boldsymbol{\Lambda}^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma}(\mathbf{I} + \boldsymbol{\Gamma}\mathbf{X}\boldsymbol{\Lambda}^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}\boldsymbol{\Gamma}\mathbf{X}\boldsymbol{\Lambda}^{-1}\mathbf{X}^{\mathrm{T}}. \quad (21)$$

The inversion $(\mathbf{I} + \boldsymbol{\Gamma}\mathbf{X}\boldsymbol{\Lambda}^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}$ requires $O(n^3)$ operations and $O(np)$ of memory (the main memory usage is for storing $\mathbf{X}$). Also, the other matrix–matrix multiplications require $O(n^2 p + n^3)$ operations. Hence, overall if we use the matrix inversion lemma, then the calculation of $\mathbf{H}$ requires $O(n^3 + n^2 p)$ operations. In summary, the calculation of $\mathbf{H}$ requires $O\{\min(p^3 + np^2, n^3 + n^2 p)\}$. Also, the amount of memory that is required by the algorithm is $O(np)$. The last step of the ALO-algorithm, i.e. step 3 in algorithm 1, requires only $O(np)$ operations. Hence, the calculation of ALO in algorithm 1 requires,

(a) through the interior point method, $O\{\min(p^3 \log(1/\epsilon) + p^3 + np^2, p^3 \log(1/\epsilon) + n^3 + n^2 p)\}$, and,
(b) through accelerated gradient descent, $O\{\min(np/\sqrt{\epsilon} + p^3 + np^2, np/\sqrt{\epsilon} + n^3 + n^2 p)\}$.

Similarly, the calculation of LO requires solving n optimization problems of the form (4). Hence, the numbers of floating point operations that are required for LO are,

(a) through the interior point method, $O\{np^3 \log(1/\epsilon)\}$, and,
(b) through accelerated gradient descent, $O(n^2 p/\sqrt{\epsilon})$.

3.2. Algorithm 2

In algorithm 2, we have used the specific form of the regularizer and simplified the form of $\mathbf{H}$. Hence, this allows for faster calculation of $\mathbf{H}$ and equivalently faster calculation of the ALO-estimate. Again the first step of calculating ALO is to solve the optimization problem. Solving this optimization problem by the interior point method or accelerated proximal gradient descent requires $O\{p^3 \log(1/\epsilon)\}$ and $O(np/\sqrt{\epsilon})$ floating point operations respectively. The next step is to calculate $\mathbf{H}$. If $\hat{\boldsymbol{\beta}}$ is s sparse, i.e. has only s non-zero coefficients, then the calculation of $\mathbf{H}$ requires $O(s^3 + ns^2)$ floating point operations. Also, the amount of memory that is required for this inversion is $O(s^2)$. Finally, the last step requires $O(np)$ operations. Hence, calculating an ALO-estimate of the risk requires,

(a) through the interior point method, $O\{p^3 \log(1/\epsilon) + s^3 + ns^2 + np\}$, and,
(b) through accelerated proximal gradient descent, $O(np/\sqrt{\epsilon} + s^3 + ns^2 + np)$.

The calculations of LO in the worst case are similar to what we had in the previous section:

(a) through the interior point method, $O\{np^3 \log(1/\epsilon)\}$, and,
(b) through accelerated proximal gradient descent, $O(n^2 p/\sqrt{\epsilon})$.

(It is known that after a finite number of iterations the estimates of proximal gradient descent become sparse, and hence the iterations require fewer operations. Hence, in practice the sparsity can reduce the computational complexity of calculating LO even though this gain is not captured in the worst-case analysis of this section.)

In this section, we used the number of floating point operations to compare the computational complexity of ALO and LO. However, since this approach is based on the worst-case scenarios and is not capable of capturing the constants, it is less accurate than comparing the timing of algorithms through simulations. Hence, Section 5 compares the performance of ALO and LO through simulations.

3.3. Memory usage

First, we discuss algorithm 1. We consider only the accelerated gradient descent algorithm. As discussed above, the amount of memory that is required for step 1 for ALO is $O(np)$ (the main memory usage is for storing matrix X). For the second step, direct inversion of $[\lambda \operatorname{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\} + \mathbf{X}^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{I}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}]^{-1}$ requires $O(p^2)$ of memory. However, by using the formula derived in equation (21) the memory usage reduces to $O(n^2)$ (for inverting $(\mathbf{I} + \boldsymbol{\Gamma}\mathbf{X}\boldsymbol{\Lambda}^{-1}\mathbf{X}^{\mathrm{T}}\boldsymbol{\Gamma})^{-1}$). Hence, the total amount of memory that is required for the second step of algorithm 1 is $O\{\min(np + n^2, np + p^2)\}$: np for storing $\mathbf{X}$ and n^2 or p^2 for calculating $[\lambda \operatorname{diag}\{\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\} + \mathbf{X}^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{I}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}]^{-1}$. The last step for ALO requires a negligible amount of memory. Hence, the total amount of memory that ALO requires, especially when $n < p$, is $O(np + n^2)$, which is the same as $O(np)$. Note that the amount of memory that is required by LO is also $O(np)$, since it requires to store $\mathbf{X}$.

The situation is even more favourable for ALO in algorithm 2; all the memory requirements are the same as before, except that the amount of memory that is required for the calculation and storing of $[\mathbf{X}_S^{\mathrm{T}} \operatorname{diag}\{\ddot{\mathbf{I}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_S + 2\lambda_1 \mathbf{I}]^{-1}$ is $O(s^2)$.

4. Theoretical results in high dimensions

4.1. Assumptions

In this section, we introduce assumptions that are later used in our theoretical results. The assumptions and theoretical results that follow are presented for finite sample sizes. However, the final conclusions of this paper are focused on the high dimensional asymptotic setting in which $n, p \to \infty$ and $n/p \to \delta_{\mathrm{O}}$, where δ_{O} is a finite number bounded away from zero. Hence,

if we write a constant as $c(n)$, it may be that the constant depends on both n and p but, since $p \sim n/\delta_{\mathrm{O}}$, we drop the dependence on p. We use this simplification for brevity and clarity of presentation. Since our major theorem involves finite sample sizes it is straightforward to go beyond this high dimensional asymptotic setting and to obtain more general results that are useful for other asymptotic settings.

Assumption 5. The rows of $\mathbf{X} \in R^{n \times p}$ are independent zero-mean Gaussian vectors with covariance $\boldsymbol{\Sigma}$. Let $\rho_{\max}$ denote the largest eigenvalue of $\boldsymbol{\Sigma}$.

As we mentioned earlier, in our asymptotic setting, we assume that $n/p \to \delta_{\mathrm{O}}$ for some δ_{O} bounded away from zero. Furthermore, we assume that the rows of $\mathbf{X}$ are scaled in a way that $\rho_{\max} = \Theta(1/n)$ to ensure that $\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta} = O_p(1)$ and $\boldsymbol{\beta}^{\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\beta} = O(1)$, assuming that each β_i is $O(1)$. Under this scaling the signal-to-noise ratio in each observation remains fixed as n and p grow. (Furthermore, under this scaling the optimal value of λ will be $O_p(1)$ (Mousavi *et al.*, 2018).) For more information on this asymptotic setting and scaling, the reader may refer to El Karoui (2018), Donoho and Montanari (2016), Donoho *et al.* (2011), Bayati and Montanari (2012), Weng *et al.* (2018) and Dobriban and Wager (2018).

Assumption 6. There are finite constants $c_1(n)$ and $c_2(n)$, and $q_n \to 0$ all functions of n, such that with probability at least $1 - q_n$ for all $i = 1, \ldots, n$

$$c_1(n) > \|\dot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\|_\infty, \tag{22}$$

$$c_2(n) > \sup_{t \in [0,1]} \frac{\|\ddot{\mathbf{l}}_{/i}\{(1-t)\hat{\boldsymbol{\beta}}_{/i} + t\hat{\boldsymbol{\beta}}\} - \ddot{\mathbf{l}}_{/i}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i} - \hat{\boldsymbol{\beta}}\|_2}, \tag{23}$$

$$c_2(n) > \sup_{t \in [0,1]} \frac{\|\ddot{\mathbf{r}}\{(1-t)\hat{\boldsymbol{\beta}}_{/i} + t\hat{\boldsymbol{\beta}}) - \ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i} - \hat{\boldsymbol{\beta}}\|_2}. \tag{24}$$

In what follows, for various regularizers and regression methods, by explicitly quantifying constants $c_1(n)$ and $c_2(n)$, we discuss conditions (22)–(24) in assumption 6. We consider the ridge regularizer in lemma 1 and the smoothed l_1- (and elastic net) regularizer in lemma 2. Concerning various regression methods, we consider logistic (lemma 3), robust regression (lemma 4), least squares (lemmas 6 and 7) and Poisson (lemmas 8 and 9) regression. The results below show that under mild assumptions, for the cases mentioned above, $c_1(n)$ and $c_2(n)$ are polynomial functions of $\log(n)$: a result that plays a key role in our main theoretical result presented in Section 4.2.

Lemma 1. For the ridge regularizer $r(z) = z^2$, we have

$$\sup_{t \in [0,1]} \frac{\|\ddot{\mathbf{r}}\{(1-t)\hat{\boldsymbol{\beta}}_{/i} + t\hat{\boldsymbol{\beta}}\} - \ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i} - \hat{\boldsymbol{\beta}}\|_2} = 0.$$

For simplicity we skip the proof. As mentioned in Section 2.2, a standard smooth approximation of the l_1-norm is given by

$$r^\alpha(z) = \sum_{i=1}^{p} \frac{1}{\alpha}[\log\{1 + \exp(\alpha z)\} + \log\{1 + \exp(-\alpha z)\}].$$

Lemma 2. For the smoothed l_1-regularizer we have

$$\sup_{t\in[0,1]} \frac{\|\ddot{\mathbf{r}}\{(1-t)\hat{\boldsymbol{\beta}}_{/i}+t\hat{\boldsymbol{\beta}}\}-\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i}-\hat{\boldsymbol{\beta}}\|_2} \leqslant 4\alpha^2.$$

We present the proof of this result in the on-line appendix section A.5.6. As a consequence of lemma 2, for the smoothed elastic net regularizer, defined as $r(z)=\gamma z^2+(1-\gamma)r^{\alpha}(z)$ for $\gamma\in[0,1]$, we have

$$\sup_{t\in[0,1]} \frac{\|\ddot{\mathbf{r}}\{(1-t)\hat{\boldsymbol{\beta}}_{/i}+t\hat{\boldsymbol{\beta}}\}-\ddot{\mathbf{r}}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i}-\hat{\boldsymbol{\beta}}\|_2} \leqslant 4(1-\gamma)\alpha^2.$$

Lemma 3. In the generalized linear model family, for the negative logistic regression log-likelihood $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})=-y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}+\log\{1+\exp(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\}$, where $y\in\{0,1\}$, we have

$$\sup_{t\in[0,1]} \frac{\|\ddot{\mathbf{I}}_{/i}\{(1-t)\hat{\boldsymbol{\beta}}_{/i}+t\hat{\boldsymbol{\beta}}\}-\ddot{\mathbf{I}}_{/i}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i}-\hat{\boldsymbol{\beta}}\|_2} \leqslant \sqrt{\sigma_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})},$$
$$\|\dot{l}(\boldsymbol{\beta})\|_{\infty}\leqslant 1.$$

We present the proof of this result in the on-line appendix section A.5.1. Our next example is about a smooth approximation of the Huber loss that is used in robust estimation, known as the pseudo-Huber-loss:

$$f_{\mathrm{H}}(z)=\gamma^2\left[\sqrt{\left\{1+\left(\frac{z}{\gamma}\right)^2\right\}}-1\right],$$

where $\gamma>0$ is a fixed number.

Lemma 4. For the pseudo-Huber-loss function $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})=f_{\mathrm{H}}(y-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$, we have

$$\sup_{t\in[0,1]} \frac{\|\ddot{\mathbf{I}}_{/i}\{(1-t)\hat{\boldsymbol{\beta}}_{/i}+t\hat{\boldsymbol{\beta}}\}-\ddot{\mathbf{I}}_{/i}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i}-\hat{\boldsymbol{\beta}}\|_2} \leqslant \frac{3}{\gamma}\sqrt{\sigma_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})},$$
$$\|\dot{l}(\boldsymbol{\beta})\|_{\infty}\leqslant \gamma.$$

The proof of this result is presented in the on-line appendix section A.5.4.

Lemma 5. If assumption 5 holds with $\rho_{\max}=c/n$, and $\delta_0=n/p$, then

$$\Pr\left\{\sigma_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})\geqslant c\left(1+3\frac{1}{\sqrt{\delta_0}}\right)^2\right\}\leqslant\exp(-p).$$

The proof of lemma 5 is presented in the on-line appendix section A. Putting together lemmas 1–5, we conclude that for ridge or smoothed l_1-regularized robust or logistic regression we have $c_1(n)=O(1)$ and $c_2(n)=O(1)$.

Lemma 6. For the loss function $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})=\frac{1}{2}(y-\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2$, we have

$$\sup_{t\in[0,1]} \frac{\|\ddot{\mathbf{I}}_{/i}\{(1-t)\hat{\boldsymbol{\beta}}_{/i}+t\hat{\boldsymbol{\beta}}\}-\ddot{\mathbf{I}}_{/i}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i}-\hat{\boldsymbol{\beta}}\|_2} = 0,$$
$$\|\dot{l}(\hat{\boldsymbol{\beta}})\|_{\infty}\leqslant \|\mathbf{y}-\mathbf{X}\hat{\boldsymbol{\beta}}\|_{\infty}.$$

We skip the proof of lemma 6 because it is straightforward.

Lemma 7. Assume that $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}^*, \sigma_\epsilon^2 \mathbf{I})$, and $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = \frac{1}{2}(y - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2$. Let assumption 5 hold with $\rho_{\max} = c/n$. Finally, let $n/p = \delta_0$ and $(1/n)\|\boldsymbol{\beta}^*\|_2^2 = \tilde{c}$. If $r(\beta) = \gamma\beta^2 + (1-\gamma)r^\alpha(\beta)$, and $0 < \gamma < 1$, then

$$\Pr\{\|\mathbf{y} - \mathbf{X}\hat{\boldsymbol{\beta}}\|_\infty > \tilde{\zeta}\sqrt{\log(n)}\} \leqslant \frac{10}{n} + 2n\exp(-n+1) + n\exp(-p),$$

where $\tilde{\zeta}$ is a constant that depends on only σ_ϵ, α, c, $\tilde{c}$, λ, δ_0 and γ (and is free of n and p).

We present the proof of this result in the on-line appendix section A.5.5. Putting together lemmas 1, 2, 6 and 7, we conclude that for smoothed elastic net regularized least squares regression we have $c_1(n) = O\{\sqrt{\log(n)}\}$ and $c_2(n) = O(1)$.

Lemma 8. In the generalized linear model family, for the negative Poisson regression log-likelihood $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = -f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) + y\log\{f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})\} - \log(y!)$ with the conditional mean $E[y|\mathbf{x}, \boldsymbol{\beta}] = f(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$ where $f(z) = \log\{1 + \exp(z)\}$ (known as a soft rectifying non-linearity), we have

$$\sup_{t\in[0,1]} \frac{\|\ddot{\mathbf{l}}_{/i}\{(1-t)\hat{\boldsymbol{\beta}}_{/i} + t\hat{\boldsymbol{\beta}}\} - \ddot{\mathbf{l}}_{/i}(\hat{\boldsymbol{\beta}})\|_2}{\|\hat{\boldsymbol{\beta}}_{/i} - \hat{\boldsymbol{\beta}}\|_2} \leqslant (1 + 6\|\mathbf{y}\|_\infty)\sqrt{\sigma_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})}$$

$$\|\dot{l}(\boldsymbol{\beta})\|_\infty \leqslant 1 + \|\mathbf{y}\|_\infty.$$

The 'soft rectifying' non-linearity $f(z) = \log\{1 + \exp(z)\}$ behaves linearly for large z and decays exponentially on its left-hand tail. Owing to the convexity and log-concavity of this non-linearity the log-likelihood is concave (Paninski, 2004), leading to a convex estimation problem. Since the actual non-linearity of neural systems is often subexponential, the soft rectifying non-linearity is popular in analysing neural data (see Pillow (2007), Park *et al.* (2014), Alison and Pillow (2017) and Zolrowski and Pillow (2018) and reference therein).

We present the proof of this result in the on-line appendix section A.5.2.

Lemma 9. Assume that $y_i \sim \text{Poisson}\{f(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*)\}$ where $f(z) = \log\{1 + \exp(z)\}$. Let assumption 5 hold with $\rho_{\max} = c/n$. Finally, let $n/p = \delta_0$ and $\boldsymbol{\beta}^{*\mathrm{T}}\boldsymbol{\Sigma}\boldsymbol{\beta}^* = \tilde{c}$. Then, for sufficiently large n, we have

$$\Pr\{(1 + 6\|\mathbf{y}\|_\infty)\sqrt{\sigma_{\max}(\mathbf{X}^{\mathrm{T}}\mathbf{X})} \geqslant \zeta_1 \log^{3/2}(n)\} \leqslant n^{1-\log\{\log(n)\}} + \frac{2}{n} + \exp\left[-n\log\left\{\frac{1}{\mathbb{P}(Z \leqslant 1)}\right\}\right] + \exp(-p)$$

$$\Pr\{\|\mathbf{y}\|_\infty \geqslant 6\sqrt{\tilde{c}}\log^{3/2}(n)\} \leqslant n^{1-\log\{\log(n)\}} + \frac{2}{n} + \exp\left[-n\log\left\{\frac{1}{\mathbb{P}(Z \leqslant 1)}\right\}\right]$$

where $Z \sim N(0, \tilde{c})$ and ζ_1 is a constant that depends on only c, $\tilde{c}$ and δ_0 (and is free of n and p).

The proof of this result is presented in the on-line appendix section A.5.3. Putting together lemmas 1, 2, 8 and 9, we conclude that for ridge or smoothed elastic net regularized Poisson regression we have $c_2(n) = O\{\log^{3/2}(n)\}$ and $c_1(n) = O\{\log^{3/2}(n)\}$.

In summary, in the high dimensional asymptotic setting, for all the examples that we have discussed so far, $c_1(n) = O\{\log^{3/2}(n)\}$ and $c_2(n) = O\{\log^{3/2}(n)\}$. Hence, in the results that we shall see in the next section we assume that both $c_1(n)$ and $c_2(n)$ are polynomial functions of $\log(n)$. Finally, we assume that the curvatures of the optimization problems that are involved in expression (1) and (4) have a lower bound.

Assumption 7. There is a constant $\nu > 0$, and a sequence $\tilde{q}_n \to 0$ such that for all $i = 1, \ldots, n$

$$\inf_{t \in [0,1]} \sigma_{\min}(\lambda \operatorname{diag}[\ddot{\mathbf{r}}\{t\hat{\boldsymbol{\beta}} + (1-t)\hat{\boldsymbol{\beta}}_{/i}\}] + \mathbf{X}_{/i}^{\mathrm{T}} \operatorname{diag}[\ddot{\mathbf{l}}_{/i}\{t\hat{\boldsymbol{\beta}} + (1-t)\hat{\boldsymbol{\beta}}_{/i}\}]\mathbf{X}_{/i}) \geqslant \nu \tag{25}$$

with probability at least $1 - \tilde{q}_n$. Here, $\sigma_{\min}(\mathbf{A})$ stands for the smallest singular value of $\mathbf{A}$.

Assumption 7 means that optimization problems (1) and (4) are strongly convex, and strong convexity is a standard assumption that is made in the analysis of high dimensional problems, e.g. Van de Geer (2008) and Negahban *et al.* (2012). Moreover, if $r(\beta) = \gamma\beta^2 + (1-\gamma)r^{\alpha}(\beta)$, and $0 < \gamma < 1$, then $\nu = 2\gamma$.

Before we mention our main result, we should also mention that assumptions 7, 5 and 6 can be weakened at the expense of making our final result look more complicated. For instance, the Gaussianity of the rows of $\mathbf{X}$ can be replaced with the sub-Gaussianity assumption with minor changes in our final result. We expect that our results (or slightly weaker results) will hold even when the rows of $\mathbf{X}$ have heavier tails. However, for brevity we do not study such matrices in the current paper. Furthermore, the smoothness of the second derivatives of the loss function and the regularizer that is assumed in expressions (23) and (24) can be weakened at the expense of slower convergence in theorem 3. We shall clarify this point in a footnote after expression (142) in the proof in the on-line appendix.

4.2. Main theoretical result

Now on the basis of these results we bound the difference $|\mathrm{ALO} - \mathrm{LO}|$. The proof is given in the on-line appendix section A.6.

Theorem 3. Let $n/p = \delta_0$ and assumption 5 hold with $\rho_{\max} = c/p$. Moreover, suppose that assumptions 6 and 7 are satisfied, and that n is sufficiently large that $q_n + \tilde{q}_n < 0.5$. Then with probability at least

$$1 - 4n\exp(-p) - \frac{8n}{p^3} - \frac{8n}{(n-1)^3} - q_n - \tilde{q}_n$$

the following bound is valid:

$$\max_{1 \leqslant i \leqslant n} \left| \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}}_{/i} - \mathbf{x}_i^{\mathrm{T}} \hat{\boldsymbol{\beta}} - \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})} \frac{H_{ii}}{1 - H_{ii}} \right| \leqslant \frac{C_{\mathrm{o}}}{\sqrt{p}}, \tag{26}$$

where

$$C_{\mathrm{o}} \triangleq \frac{72c^{3/2}}{\nu^3} \left\{ 1 + \sqrt{\delta_0}(\sqrt{\delta_0} + 3)^2 \frac{c\log(n)}{\log(p)} \right\} \left[c_1^2(n)c_2(n) + c_1^3(n)c_2^2(n) \frac{5\{c^{1/2} + c^{3/2}(\sqrt{\delta_0} + 3)^2\}}{\nu^2} \right]. \tag{27}$$

Recall that in Section 4.1 we proved that for many regularized regression problems in the generalized linear family both $c_1(n) = O\{\mathrm{PolyLog}(n)\}$ and $c_2(n) = O\{\mathrm{PolyLog}(n)\}$, where the notation $\mathrm{PolyLog}(n)$ denotes a polynomial in $\log(n)$. These examples included ridge and smoothed l_1- (and elastic net) regularizers and logistic, robust, least squares and Poisson regression. More specifically, the maximum degree that we observed for the logarithm was $\frac{3}{2}$, which happened for Poisson regression. Furthermore, as mentioned in the previous section, in the high dimensional asymptotic setting in which $n, p \to \infty$ and $n/p \to \delta_{\mathrm{o}}$, where δ_{o} is a finite number bounded away from zero, to keep the signal-to-noise ratio fixed in each observation (as p and n grow),

we considered the scaling that $n\rho_{\max} = O(1)$. Combining these, it is straightforward to see that $C_0(n) = O\{c_1^3(n)c_2^2(n)\} = O\{\text{PolyLog}(n)\}$. Therefore, the difference

$$\max_{1\leqslant i\leqslant n}\left|\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i} - \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} - \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})}\frac{H_{ii}}{1-H_{ii}}\right| = O_p\left\{\frac{\text{PolyLog}(n)}{\sqrt{n}}\right\}.$$

Theorem 3 proves the accuracy of the approximation of the leave-one-out estimate of the regression coefficients. As a simple corollary of this result we can also prove the accuracy of our approximation of LO.

Corollary 1. Suppose that all the assumptions that are used in theorem 3 hold. Moreover, suppose that

$$\max_{i=1,2,\ldots,n}\sup_{|b_i|<C_{\mathrm{o}}/\sqrt{p}}|\dot{\phi}(y_i, \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i} + b_i)| \leqslant c_3(n)$$

with probability r_n. Then, with probability at least

$$1 - 4n\exp(-p) - \frac{8n}{p^3} - \frac{8n}{(n-1)^3} - q_n - \tilde{q}_n - r_n,$$

$$|\text{ALO} - \text{LO}| \leqslant \frac{c_3(n)C_{\mathrm{o}}}{\sqrt{p}}, \tag{28}$$

where C_{o} is the constant that is defined in theorem 3.

The proof of this result can be found in the on-line appendix section A.8. As we discussed before, in all the examples that we have seen so far $C_{\mathrm{o}}/\sqrt{p}$ is $O\{\text{PolyLog}(n)/\sqrt{n}\}$. Hence, to obtain the convergence rate of ALO to LO we need to find only an upper bound for $c_3(n)$. Note that usually the loss function l that is used in the optimization problem is also used as the function ϕ to measure the prediction error. Hence, assuming that $\phi(\cdot,\cdot) = l(\cdot,\cdot)$, we study the value of $c_3(n)$ for the examples that we discussed in Section 4.1.

(a) If ϕ is the loss function of lemma 3, then $|\dot{\phi}(y_i, \mathbf{x}_i^{\mathrm{T}}\beta)| \leqslant 2$, leading to $c_3(n) = 2$.
(b) If ϕ is the loss function of lemma 8, then $|\dot{\phi}(y_i, \mathbf{x}_i^{\mathrm{T}}\beta)| \leqslant 1 + \|\mathbf{y}\|_\infty$. Furthermore, we proved in lemma 9 that, under the data-generating mechanism that was described there, with high probability $\|\mathbf{y}\|_\infty < 6\sqrt{\{\tilde{c}\log^3(n)\}}$, leading to $c_3(n) = 1 + 6\sqrt{\{\tilde{c}\log^3(n)\}}$.
(c) For the pseudo-Huber-loss that is described in lemma 4, we have $|\dot{\phi}(y_i, \mathbf{x}_i^{\mathrm{T}}\beta)| \leqslant \gamma$, leading to $c_3(n) = \gamma$.
(d) For the square loss

$$|\dot{\phi}(y_i, \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i} + b_i)| \leqslant |y_i - \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i}| + |b_i| \leqslant |y_i - \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i}| + \frac{C_{\mathrm{o}}}{\sqrt{p}}.$$

Hence, to obtain a proper upper bound we require more information about the estimate $\hat{\boldsymbol{\beta}}_{/i}$. Suppose that our estimates are obtained from the optimization problem that we discussed in lemma 7. Then, on the basis of expressions (94) and (97) in the proof of lemma 7 in the on-line appendix A.5.5

$$\max_i |y_i - \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i}| \leqslant \max_i |y_i| + \max_i |\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}_{/i}| \leqslant 2\sqrt{\{(c\tilde{c} + \sigma_\epsilon^2)\log(n)\}} + \sqrt{\left\{\frac{10c(c\tilde{c} + \sigma_\epsilon^2)\log(n)}{\lambda\gamma}\right\}}$$

with probability at most $4/n + n\exp(-n+1)$, leading to

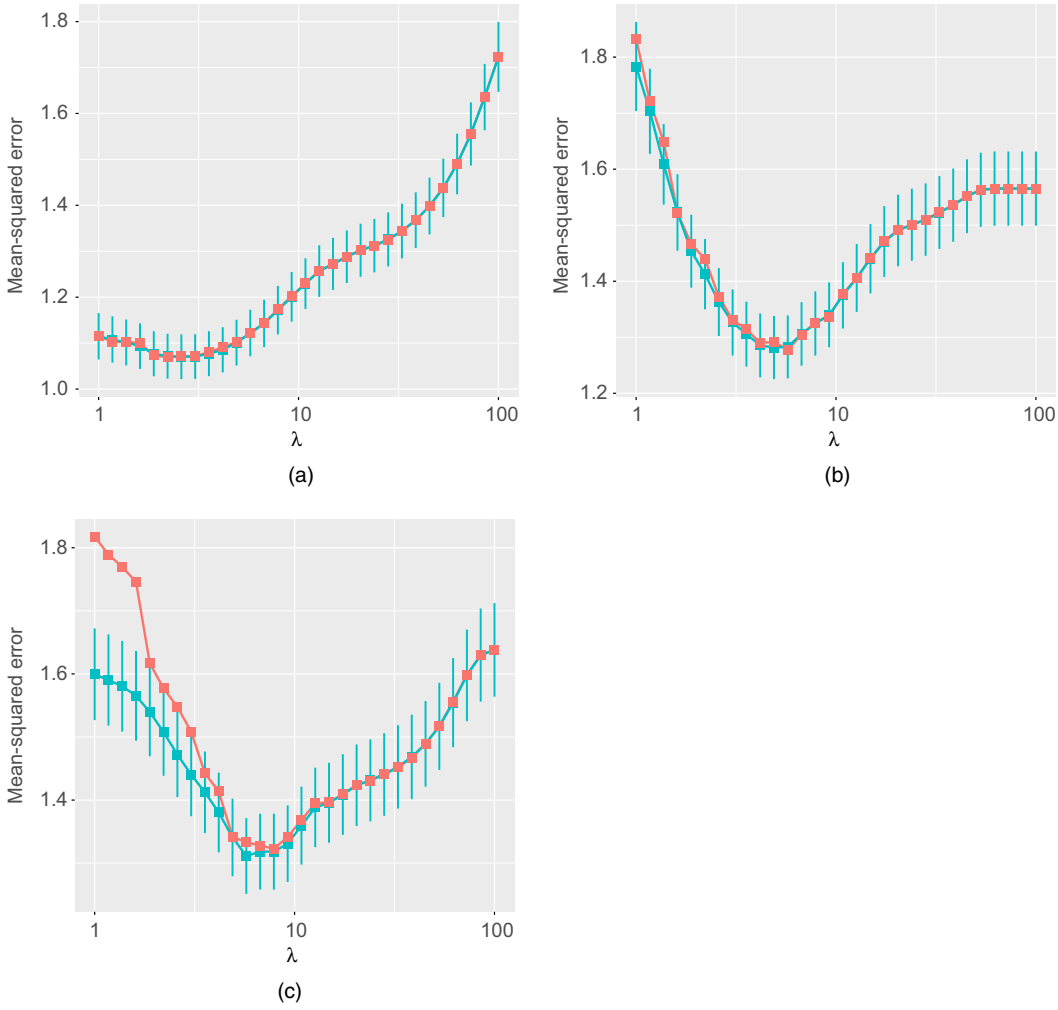

Fig. 4. ALO (■) and LO (■) mean-square error for elastic net linear regression |, 1 standard error interval of LO): (a) $n > p$ ($n = 1000$; $p = 200$; LO, 40.47 s; ALO, 0.30 s; fit, 0.03 s); (b) $n = p$ ($n = 1000$; $p = 1000$; LO, 360.46 s; ALO, 4.72 s; fit, 0.33 s); (c) $n < p$ ($n = 1000$; $p = 10000$; LO, 1377.37 s; ALO, 31.14 s; fit, 1.23 s)

$$c_3(n) = 2\sqrt{\{(c\tilde{c} + \sigma_\epsilon^2)\log(n)\}} + \sqrt{\left\{\frac{20c(c\tilde{c} + \sigma_\epsilon^2)\log(n)}{\lambda\gamma}\right\}} + \frac{C_0}{\sqrt{p}}.$$

In summary, in the high dimensional asymptotic setting, for the regularized regression methods that were introduced in Section 4.1, such as least squares, logistic, Poisson and robust regression, with $r(\beta) = \gamma\beta^2 + (1-\gamma)r^\alpha(\beta)$, and $0 < \gamma < 1$, and assuming that $\phi(\cdot,\cdot) = l(\cdot,\cdot)$, we have that $c_3(n) = O\{\text{PolyLog}(n)\}$, leading to $|\text{ALO} - \text{LO}| = O_p\{\text{PolyLog}(n)/\sqrt{n}\}$. In short, these examples show that ALO offers a consistent estimate of LO.

Finally, note that in the p fixed, $n \to \infty$ regime, theorem 3 fails to yield $|\text{ALO} - \text{LO}| = o_p(1)$. This is just an artefact of our proof. In theorem 6, which is presented in the on-line appendix section A.9, we prove that, under mild regularity conditions, the error between ALO and LO is $o_p(1/n)$ when $n \to \infty$ and p is fixed. For brevity details are presented in the on-line appendix section A.9.

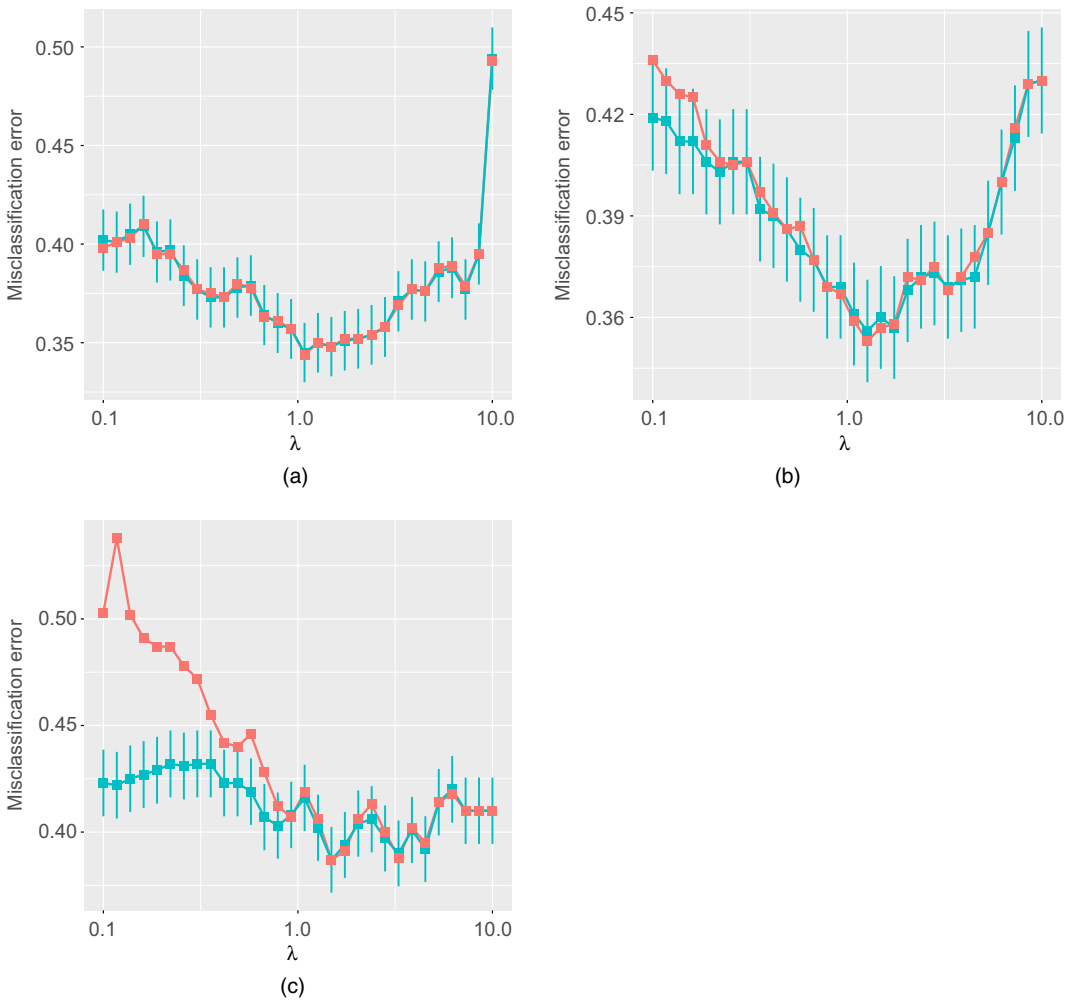

Fig. 5. ALO (■) and LO (■) misclassification errors (as a function of λ) for lasso logistic regression (|, 1 standard error interval of LO): (a) $n > p$ ($n = 1000$; $p = 200$; LO, 148.40 s; ALO, 0.16 s; fit, 0.14 s); (b) $n = p$ ($n = 1000$; $p = 1000$; LO, 960.41 s; ALO, 1.02 s; fit, 0.89 s); (c) $n < p$ ($n = 1000$; $p = 10000$; LO, 1525.76 s; ALO, 1.87 s; fit, 1.46 s)

5. Numerical experiments

5.1. Summary

To illustrate the accuracy and computational efficiency of approximate leave-one-out cross-validation we apply it to synthetic and real data. We generate synthetic data, and compare ALO and LO for elastic net linear regression in Section 5.2.1, lasso logistic regression in Section 5.2.2, and elastic net Poisson regression in Section 5.2.3. We emphasize that our simulations were performed on a single personal computer, and we have not considered the effect of parallelization on the performance of ALO and LO. In other words, the simulation results that are reported for LO are based on its sequential implementation on a single personal computer. For real data, we apply lasso, elastic net and ridge logistic regression to sonar returns from two undersea targets in Section 5.3.1, and we apply lasso Poisson regression to real recordings from spatially sensitive

neurons in Section 5.3.2. Our synthetic and real data examples cover various data shapes where $n > p$, $n = p$ and $n < p$.

Figs 4, 5, 6 and 7 and Fig. 8(e) reveal that ALO offers a reasonably accurate estimate of LO for a large range of λ. These figures show that ALO deteriorates for extremely small values of λ, especially when $p > n$. This is not a serious issue because the λs minimizing LO and ALO tend to be far from those small values.

The real data example in Section 5.3.1, illustrating ALO and LO in Fig. 7, is about classifying sonar returns from two undersea targets by using penalized logistic regression. The neuroscience example in Section 5.3.2 is about estimating an inhomogeneous spatial point process by using an overcomplete basis from a sparsely sampled two-dimensional space. Given the spatial nature of the problem, the design matrix $\mathbf{X}$ is very sparse, which fails to satisfy the dense Gaussian design assumption that we made in theorem 3. Nevertheless, Fig. 8(e) illustrates the excellent performance of approximate leave-one-out cross-validation in approximating LO in an example where $p = 10000$ and $n = 3133$.

Fig. 2 compares the computational complexity (time) of a single fit, ALO and LO, as we increase p while we keep the ratio n/p fixed. We consider various data shapes, models and penalties. Fig. 2(a) shows time *versus* p for elastic net linear regression when $n/p = 5$. Fig. 2(b) shows time *versus* p for lasso logistic regression when $n/p = 1$. Fig. 2(c) shows time *versus* p for elastic net Poisson regression when $n/p = \frac{1}{10}$. Finally, Fig. 8(e) shows that for the neuroscience example approximate leave-one-out cross-validation takes 7 s in comparison with the 60428 s that are required by leave-one-out cross-validation. All these numerical experiments illustrate the significant computational saving that is offered by approximate leave-one-out cross-validation. As it pertains to the reported run times, all fits in this paper were performed using a 3.1-GHz Intel Core i7 MacBook Pro with 16 Gbytes of memory. All the code for the figures that are presented in this paper are available from `https://github.com/RahnamaRad/ALO`.

5.2. Simulations

In all the examples in this section (Sections 5.2.1, 5.2.2, 5.2.3 and 5.2.4), we let the true unknown parameter vector $\boldsymbol{\beta}^* \in R^p$ have $k = n/10$ non-zero coefficients. The k non-zero coefficients are randomly selected, and their values are independently drawn from a zero-mean unit variance Laplace distribution. The rows $\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_n^{\mathrm{T}}$ of the design matrix $\mathbf{X}$ are independently drawn from $N(0, \boldsymbol{\Sigma})$. We consider two correlation structures:

(a) *spiked*, $\mathrm{corr}(X_{ij}, X_{ij'}) = 0.5$, and
(b) *Toeplitz*, $\mathrm{corr}(X_{ij}, X_{ij'}) = 0.9^{|j'-j|}$.

$\boldsymbol{\Sigma}$ is scaled such that the signal variance $\mathrm{var}(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*) = 1$ regardless of the problem dimension. In this section, all the fits and calculations of LO (and the 1-standard-error interval of LO) were computed by using the `glmnet` package in R (Friedman *et al.*, 2010), and ALO was computed by using the `alocv` package in R (He *et al.*, 2018).

5.2.1. Linear regression with elastic net penalty

We set $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = \frac{1}{2}(y - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2$, $r(\boldsymbol{\beta}) = \{(1-\alpha)/2\}\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1$ and $\alpha = 0.5$. We let the rows $\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_n^{\mathrm{T}}$ of $\mathbf{X}$ have a spiked covariance and, to generate data, we sample $\mathbf{y} \sim N(\mathbf{X}\boldsymbol{\beta}^*, \mathbf{I})$. Moreover, $\phi(y, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = (y - \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})^2$ so that

$$\mathrm{ALO} = \frac{1}{n}\sum_{i=1}^{n}\left(\frac{y_i - \mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}}{1 - H_{ii}}\right)^2$$

with $\mathbf{H} = \mathbf{X}_S\{\mathbf{X}_S^{\mathrm{T}}\mathbf{X}_S + \lambda(1-\alpha)\mathbf{I}\}^{-1}\mathbf{X}_S^{\mathrm{T}}$. For various data shapes, i.e. $n/p \in \{5,\ 1,\ \frac{1}{10}\}$, we depict

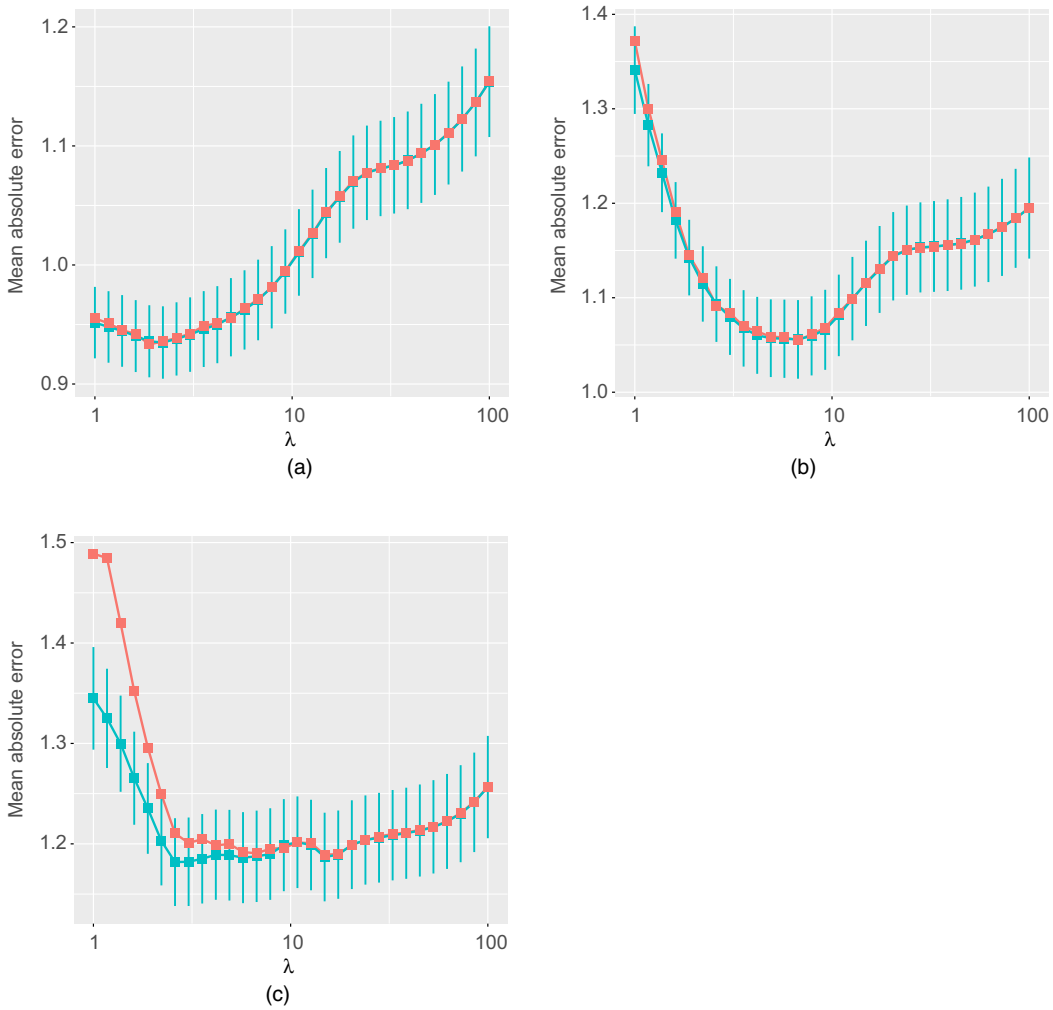

Fig. 6. ALO (■) and LO (■) mean absolute errors (as a function of λ) for elastic net Poisson regression (|, 1-standard-error interval of LO): (a) $n > p$ ($n = 1000$; $p = 200$; LO, 214.48 s; ALO, 0.78 s; fit, 0.21 s); (b) $n = p$ ($n = 1000$; $p = 1000$; LO, 830.85 s; ALO, 6.84 s; fit, 0.81 s); (c) $n < p$ ($n = 1000$; $p = 10000$; LO, 3733.53 s; ALO, 41.52 s; fit, 3.55 s)

results in Fig. 4 where reported times refer to the required time to fit the model, to compute ALO and LO for a sequence of 30 logarithmically spaced tuning parameters from 1 to 100.

5.2.2. Logistic regression with lasso penalty

We set $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = -y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} + \log\{1 + \exp(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}\}$ (the negative logistic log-likelihood) and $r(\boldsymbol{\beta}) = \|\boldsymbol{\beta}\|_1$. We let the rows $\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_n^{\mathrm{T}}$ of $\mathbf{X}$ have a *Toeplitz* covariance and, to generate data, we sample

$$y_i \sim \text{binomial}\left\{\frac{\exp(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*)}{1 + \exp(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*)}\right\}.$$

We take the misclassification rate as our measure of error, and $\mathbf{1}_{\{\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}>0\}}$ as prediction, where

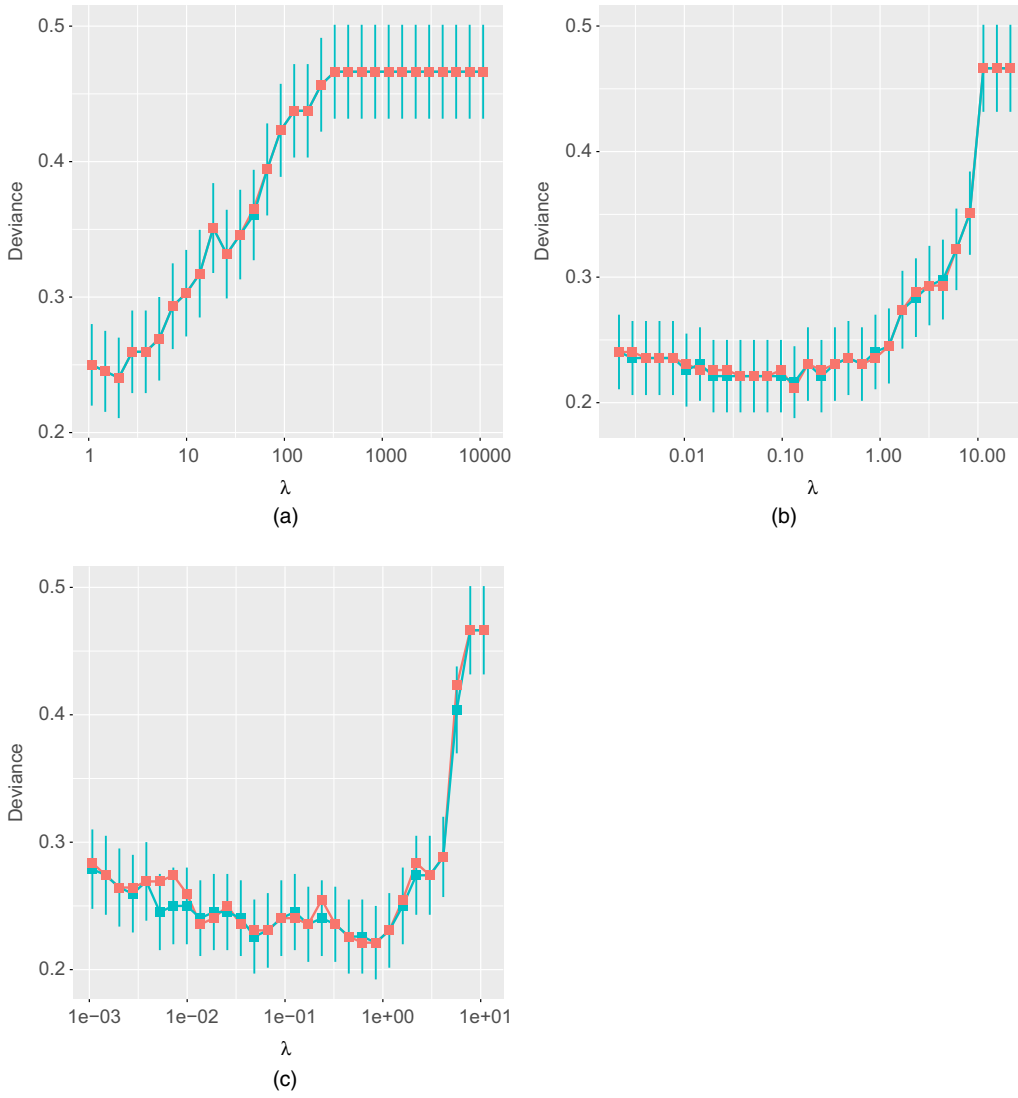

Fig. 7. ALO- (■) and LO- (■) deviances (as a function of λ) for penalized logistic regression applied to the sonar data (Section 5.3.1) where $n = 208$ and $p = 60$ (|, 1-standard-error interval of LO): (a) ridge regression (LO, 4.217 s; ALO, 0.062 s; fit, 0.016); (b) elastic net (LO, 46.884 s; ALO, 0.216 s; fit, 0.193 s); (c) lasso (LO, 129.466 s; ALO, 0.593 s; fit, 0.568 s)

$\mathbf{1}_{\{\cdot\}}$ is the indicator function, so that

$$\text{ALO} = \frac{1}{n}\sum_{i=1}^{n} |y_i - \mathbf{1}_{\{\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \{\dot{l}_i(\hat{\boldsymbol{\beta}})/\ddot{l}_i(\hat{\boldsymbol{\beta}})\}H_{ii}/(1-H_{ii})>0\}}|$$

where

$$\mathbf{H} = \mathbf{X}_S[\mathbf{X}_S^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_S]^{-1}\mathbf{X}_S^{\mathrm{T}}\mathrm{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\},$$

$$\dot{l}_i(\hat{\boldsymbol{\beta}}) = \{1 + \exp(-\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}})^{-1} - y_i$$

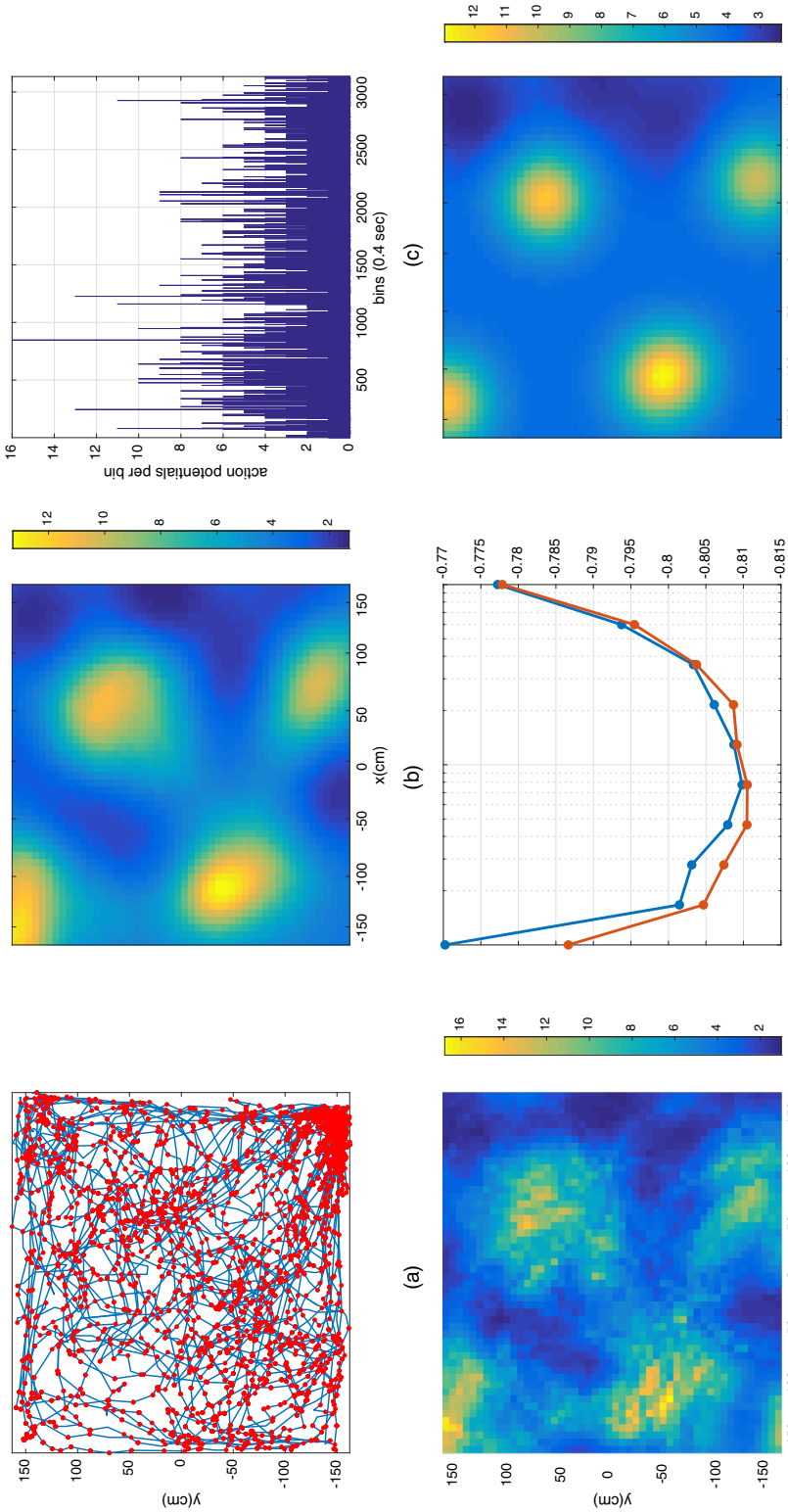

Fig. 8. (a) Spike locations (●) superimposed on the animal's trajectory (——) (firing fields are areas covered by a cluster of action potentials; the firing fields of a grid cell form a periodic triangular matrix tiling the entire environment that is available to the animal), (b) ALO-based firing rate, (c) LO-based firing rate, (d) $\lambda = 0.001$-based firing rate, (d) ALO (●, time = 7 s) and LO (●, time = 60428 s) over a wide range of λs and (e) $\lambda = 0.1$-based firing rate

and

$$\ddot{l}_i(\hat{\boldsymbol{\beta}}) = \exp(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}})\{1+\exp(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}})\}^{-2}.$$

For various data shapes, i.e. $n/p \in \{5,\ 1,\ \frac{1}{10}\}$, we depict results in Fig. 5 where reported times refer to the required time to fit the model, to compute ALO and LO for a sequence of 30 logarithmically spaced tuning parameters from 0.1 to 10.

5.2.3. Poisson regression with elastic net penalty

We set $l(y|\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = \exp(y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) - y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}$ (the negative Poisson log-likelihood),

$$r(\boldsymbol{\beta}) = \{(1-\alpha)/2\}\|\boldsymbol{\beta}\|_2^2 + \alpha\|\boldsymbol{\beta}\|_1$$

and $\alpha = 0.5$. We let the rows $\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_n^{\mathrm{T}}$ of $\mathbf{X}$ have a *spiked* covariance and, to generate data, we sample $y_i \sim \text{Poisson}\{\exp(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}^*)\}$. We use the mean absolute error as our measure of error, and $\exp(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta})$ as prediction, so that

$$\text{ALO} = \frac{1}{n}\sum_{i=1}^{n}\left| y_i - \exp\left\{\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}} + \frac{\dot{l}_i(\hat{\boldsymbol{\beta}})}{\ddot{l}_i(\hat{\boldsymbol{\beta}})}\frac{H_{ii}}{1-H_{ii}}\right\}\right|$$

where $\mathbf{H} = \mathbf{X}_S\{\mathbf{X}_S^{\mathrm{T}}\text{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}\mathbf{X}_S + \lambda(1-\alpha)\mathbf{I}\}^{-1}\mathbf{X}_S^{\mathrm{T}}\text{diag}\{\ddot{\mathbf{l}}(\hat{\boldsymbol{\beta}})\}$, $\dot{l}_i(\hat{\boldsymbol{\beta}}) = \exp(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}}) - y_i$ and $\ddot{l}_i(\hat{\boldsymbol{\beta}}) = \exp(\mathbf{x}_i^{\mathrm{T}}\hat{\boldsymbol{\beta}})$. For various data shapes, i.e. $n/p \in \{5,\ 1,\ \frac{1}{10}\}$, we depict results in Fig. 6 where reported times refer to the required time to fit the model, to compute ALO and LO for a sequence of 30 logarithmically spaced tuning parameters from 1 to 100.

5.2.4. Timing simulations

To compare the timing of ALO with that of LO, we consider the following scenarios:

(a) elastic net linear regression, with rows of the design matrix having a spiked covariance, data generated as described in Sections 5.2 and 5.2.1, and considered for a sequence of 10 logarithmically spaced tuning parameters from 1 to 100; we let $n/p = 5$;
(b) lasso logistic regression, with rows of the design matrix having a Toeplitz covariance, data generated as described in Sections 5.2 and 5.2.2, and considered for a sequence of 10 logarithmically spaced tuning parameters from 0.1 to 10; we let $n/p = 1$;
(c) elastic net Poisson regression, with rows of the design matrix having a spiked covariance, data generated as described in Sections 5.2 and 5.2.3, and considered for a sequence of 10 logarithmically spaced tuning parameters from 1 to 100; we let $n/p = \frac{1}{10}$.

The timings of a single fit, ALO and LO *versus* model complexity p are illustrated in Fig. 2. The reported timings are obtained by recording the time that was required to find a single fit and LO by using the `glmnet` package in R (Friedman *et al.*, 2010), and to find ALO by using the `alocv` package in R (He *et al.*, 2018), all along the tuning parameters above. This process is repeated five times to obtain the average timing.

5.3. Real data

5.3.1. Sonar data

Here we use ridge, elastic net and lasso logistic regression to classify sonar returns collected from a metal cylinder and a cylindrically shaped rock on a sandy ocean floor. The data consist of a set of $n = 208$ returns, 111 cylinder returns and 97 rock returns, and $p = 60$ spectral features extracted from the returning signals (Gorman and Sejnowski, 1988). We use the misclassification rate as our measure of error. Numerical results comparing ALO and LO for ridge, elastic net and

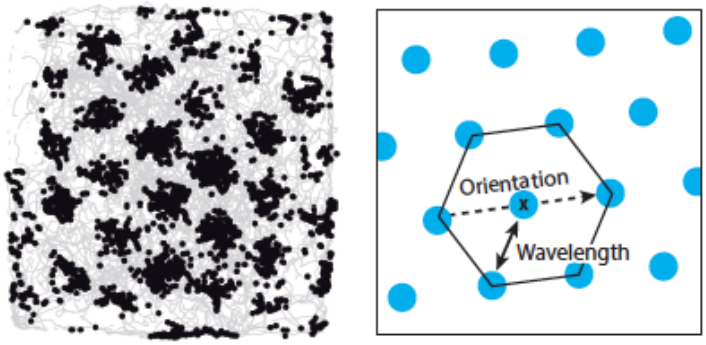

Fig. 9. (a) Spike locations (●) superimposed on an animal's trajectory (●) (firing fields are areas covered by a cluster of action potentials) and (b) the firing fields of a grid cell form a periodic triangular matrix tiling the entire environment available to the animal: the figure is adapted from Moser *et al.* (2014)

lasso logistic regression are depicted in Fig. 7. The single fit and LO (and the 1-standard-error interval of LO) were computed by using the `glmnet` package in R (Friedman *et al.*, 2010), and ALO was computed by using the `alocv` package in R (He *et al.*, 2018). The values of the tuning parameters are a sequence of 30 logarithmically spaced tuning parameters between two values automatically selected by the `glmnet` package.

5.3.2. Spatial point process smoothing of grid cells: a neuroscience application

In this section, we compare ALO with LO on a real data set. This data set includes electrical recordings of single neurons in the entorhinal cortex: an area in the brain that has been found to be particularly responsible for the navigation and perception of space in mammals (Moser *et al.*, 2008). The entorhinal cortex is also one of the areas that is pathologically affected in the early stages of Alzheimer's disease, causing symptoms of spatial disorientation (Khan *et al.*, 2014). Moreover, the entorhinal cortex provides input to another area, the hippocampus, which is involved in the cognition of space and the formation of episodic memory (Buzsaki and Moser, 2013).

Electrical recordings of single neurons in the medial domain of the entorhinal cortex of freely moving rodents have revealed spatially modulated neurons, called grid cells, firing action potentials only around the vertices of two-dimensional hexagonal lattices covering the environment in which the animal navigates. The hexagonal firing pattern of a single grid cell is illustrated in Fig. 9(a). These grid cells can be categorized according to the orientation of their triangular grid, the wavelength (distance between the vertices) and the phase (shift of the whole lattice). See Fig. 9(b) for an illustration of the orientation and wavelength of a single grid cell.

The data that we analyse here consist of extra cellular recordings of several grid cells, and the simultaneously recorded location of the rat within a 300 cm × 300 cm box for roughly 20 min. (The source of the data is Stensola *et al.* (2012). For a video of a single grid cell recorded in the medical entorhinal cortex see the clip `https://www.youtube.com/watch?v=i9GiLBXWAHI`.) Since the number of spikes that are fired by a grid cell depends mainly on the location of the animal, regardless of the animal's speed and running direction (Hafting *et al.*, 2005), it is reasonable to summarize this spatial dependence in terms of a rate map $\eta(\mathbf{r})$, where $\eta(\mathbf{r})\mathrm{d}t$ is the expected number of spikes emitted by the grid cell in a fixed time interval $\mathrm{d}t$, given that the animal is at position $\mathbf{r}$ during this time interval (Rahnama Rad and Paninski, 2010; Pnevmatikakis *et al.*,

2014; Dunn *et al.*, 2015). In other words, if the rat passes the same location again, we again expect the grid cell to fire at virtually the same rate, specifically according to a Poisson distribution with mean $\eta(\mathbf{r})\mathrm{d}t$. (It is known that these rate maps can in some cases change with time but in most cases it is reasonable to assume that they are constant. Moreover, the two-dimensional surface that is represented by $\eta(\mathbf{r})$ is not the same for different grid cells.) For each grid cell, the estimation of the rate map $\eta(\mathbf{r})$ is a first step towards understanding the cortical circuitry underlying spatial cognition (Rowland *et al.*, 2016). Consequently, the estimation of firing fields without contamination from measurement noise or bias from oversmoothing will help to clarify important questions about neuronal algorithms underlying navigation in real and mental spaces (Buzsaki and Moser, 2013).

To be concrete, we discretize the two-dimensional space into an $m \times m$ grid, and we discretize time into bins with width $\mathrm{d}t$. In this example, $\mathrm{d}t$ is 0.4 s and m is 50. The experiment is 1252.9 s long, and therefore we have $\lceil 1252.9/0.4 \rceil = 3133$ time bins. In other words, $n = 3133$. We use $y_i \in \{0, 1, 2, 3, \ldots\}$ to denote the number of action potentials that are observed in time interval $[(i-1)\mathrm{d}t, i\mathrm{d}t)$, where $i = 1, \ldots, n$. Moreover, we use $\mathbf{r}_i \in R^{m^2}$ to denote a vector composed of 0s except for a single 1 at the entry corresponding to the animal's location within the $m \times m$ grid during the time interval $[(i-1)\mathrm{d}t, i\mathrm{d}t)$. We assume a log-linear model $\log\{\eta(\mathbf{r})\} = \mathbf{r}^{\mathrm{T}}\mathbf{z}$, relating the firing rate at location $\mathbf{r} \in R^{m^2}$ to the latent vector $\mathbf{z}$ where the $m \times m$ latent spatial process that is responsible for the observed spiking activity is unravelled into $\mathbf{z} \in R^{m^2}$. The firing rate can be written as $\eta(\mathbf{r}_i) = \exp(\mathbf{r}_i^{\mathrm{T}}\mathbf{z})$. With this notation, $\mathbf{r}_i^{\mathrm{T}}\mathbf{z}$ is the value of $\mathbf{z}$ at the animal's location during the time interval $[(i-1)\mathrm{d}t, i\mathrm{d}t)$. In this vein, the distribution of observed spiking activity can be written as

$$p(y_i|\mathbf{r}_i) = \frac{\exp\{-\eta(\mathbf{r}_i)\}\eta(\mathbf{r}_i)^{y_i}}{y_i!}. \tag{29}$$

As mentioned earlier, the main goal is to estimate the two-dimensional rate map $\eta(\cdot)$, and a large body of work has addressed the problem of estimating a smooth rate map from neural data (DiMatteo *et al.*, 2001; Gao *et al.*, 2002; Kass *et al.*, 2005; Cunningham *et al.*, 2008, 2009; Czanner *et al.*, 2008; Paninski *et al.*, 2010; Rahnama Rad and Paninski, 2010; Macke *et al.*, 2011; Pnevmatikakis *et al.*, 2014). Here we employ an overcomplete basis to account for the spatially localized sensitivity of grid cells. Since it is known that the rate map of any single grid cell consists of bumps of elevated firing rates at various points in the two-dimensional space, as illustrated in Fig. 9(a), it is reasonable to represent $\mathbf{z}$ as a linear combination of $\{\psi_1, \ldots, \psi_p\}$: an overcomplete basis in R^p (Brown *et al.*, 2001; Pnevmatikakis *et al.*, 2014; Dunn *et al.*, 2015). We compose the overcomplete basis by using truncated Gaussian bumps with various scales, distributed at all pixels. The four basic Gaussian bumps that we use are depicted in Fig 10. Since we use four truncated Gaussian bumps for each pixel, in this example, we have a total of $p = 4m^2 = 10000$ basis functions. We employ the truncated Gaussian bumps

$$\exp\Big\{-\frac{1}{2\sigma^2}(u_x^2 + u_y^2)\Big\}\mathbf{1}_{\{\exp[-\{1/(2\sigma^2)\}(u_x^2+u_y^2)]>0.05\}}$$

where u_x and u_y are the horizontal and vertical co-ordinates. Define $\boldsymbol{\Psi} \in R^{m^2 \times p}$ as a matrix composed of columns $\{\psi_1, \ldots, \psi_p\}$. Furthermore, define $\tilde{\mathbf{x}}_i \in R^p$ as $\tilde{\mathbf{x}}_i =^{\triangle} \boldsymbol{\Psi}^{\mathrm{T}}\mathbf{r}_i$, and define $\tilde{\mathbf{X}} \in R^{n \times p}$ as a matrix composed of rows $\{\tilde{\mathbf{x}}_1^{\mathrm{T}}, \ldots, \tilde{\mathbf{x}}_n^{\mathrm{T}}\}$. We normalize the columns of $\tilde{\mathbf{X}}$, calling the resulting matrix $\mathbf{X}$. The columns of $\mathbf{X} \in R^{n \times p}$ are unit normed. Formally, $\mathbf{X} = \tilde{\mathbf{X}}\boldsymbol{\Gamma}^{-1}$ where $\boldsymbol{\Gamma} \in R^{p \times p}$ is a diagonal matrix filled with the column norms of $\tilde{\mathbf{X}}$. We use $\{\mathbf{x}_1^{\mathrm{T}}, \ldots, \mathbf{x}_n^{\mathrm{T}}\}$ to refer to the rows of $\mathbf{X}$, yielding $\eta(\mathbf{r}_i) = \exp(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta})$. Because of the above-mentioned rescaling, we have the following relationship between the latent map $\mathbf{z}$ and $\boldsymbol{\beta}$: $\mathbf{z} = \boldsymbol{\Psi}\boldsymbol{\Gamma}\boldsymbol{\beta}$. Sparsity of $\boldsymbol{\beta}$ refers to our

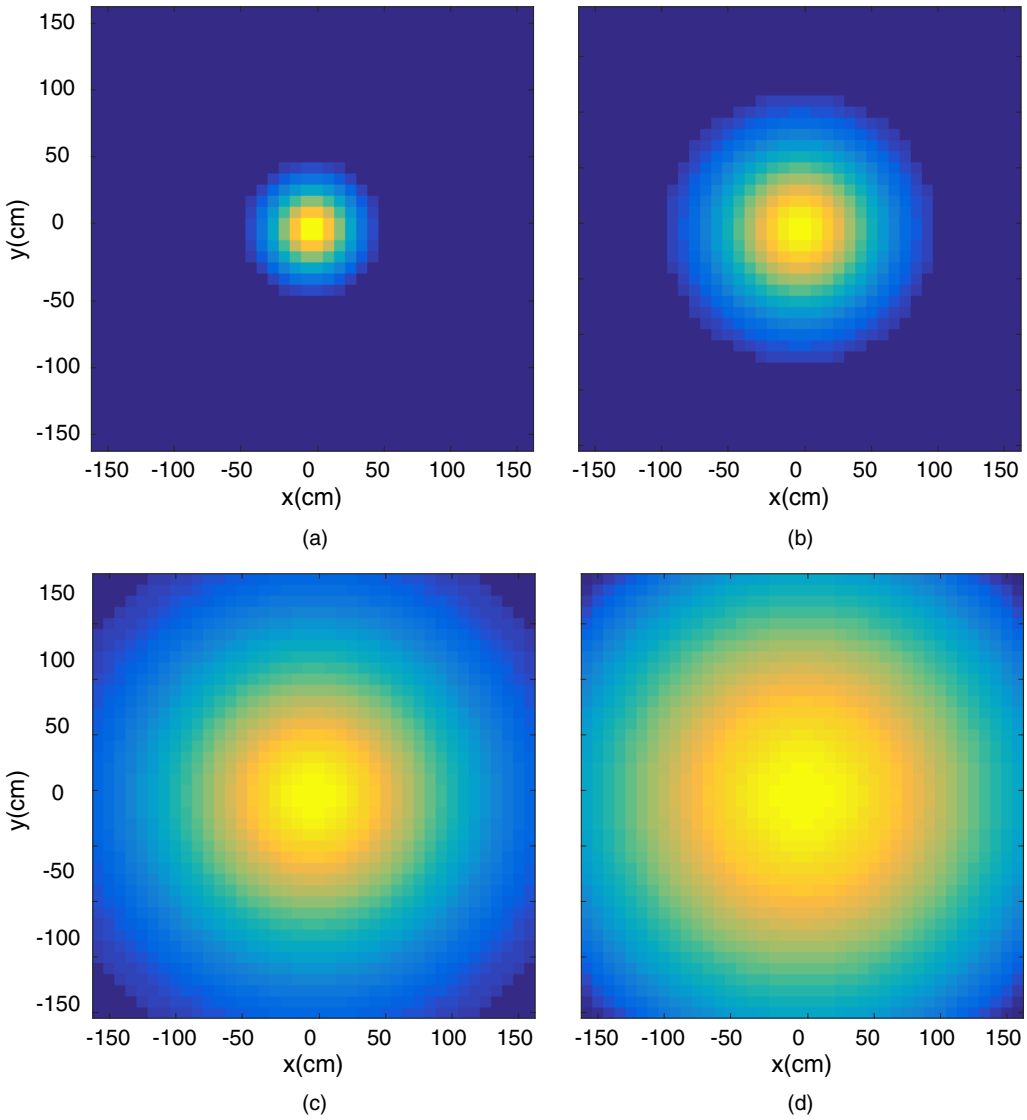

Fig. 10. The four truncated Gaussian bumps

prior understanding that the rate map of grid cells consists of bumps of elevated firing rates, at various points in the two-dimensional space and, therefore, our estimation problem is

$$\hat{\boldsymbol{\beta}} \triangleq \arg\min_{\boldsymbol{\beta}\in R^p} \left\{ \sum_{i=1}^{n} [\eta(\mathbf{r}_i) - y_i \log\{\eta(\mathbf{r}_i)\}] + \lambda\|\boldsymbol{\beta}\|_1 \right\}$$
$$= \arg\min_{\boldsymbol{\beta}\in R^p} \left[\sum_{i=1}^{n} \{\exp(\mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}) - y_i \mathbf{x}_i^{\mathrm{T}}\boldsymbol{\beta}\} + \lambda\|\boldsymbol{\beta}\|_1 \right].$$

Here we use the negative log-likelihood in equation (29) as the cost function, i.e. $\phi(y, \mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) = y\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta} - \exp(\mathbf{x}^{\mathrm{T}}\boldsymbol{\beta}) + \log(y!)$. We remind the reader that we use the ALO-formula that was ob-

tained in theorem 1. Fig. 8 illustrates that ALO is a reasonable approximation of LO, allowing computationally efficient tuning of λ. To see the effect of λ of the rate map, we also present the maps resulting from small and large values of λ, leading to undersmooth and oversmooth rate maps respectively. As it pertains to the reported run times, all fittings in this section were performed by using the `glmnet` package (Qian *et al.*, 2013) in MATLAB.

6. Concluding remarks

Leave-one-out cross-validation is an intuitive and conceptually simple risk estimation technique. Despite its low bias in estimating the extrasample prediction error, the high computational complexity of leave-one-out cross-validation has limited its applications for high dimensional problems. In this paper, by combining a single step of the Newton method with low rank matrix identities, we obtained an approximate formula for LO, called ALO. We showed how ALO can be applied to popular non-differentiable regularizers, such as the lasso. With the aid of theoretical results and numerical experiments, we showed that ALO offers a computationally efficient and statistically accurate estimate of the extrasample prediction error in high dimensions.

Important directions for future work involve various approximations that further reduce the computational complexity. The computational bottleneck of approximate leave-one-out cross-validation is the inversion of the large generalized hat matrix $\mathbf{H}$. This can make the application of approximate leave-one-out cross-validation to ultrahigh dimensional problems computationally challenging. Since the diagonals of our $\mathbf{H}$-matrix can be represented as leverage scores of an augmented $\mathbf{X}$-matrix, scalable methods to compute the leverage score approximately may offer a promising avenue for future work. For example Drineas *et al.* (2012) offered a randomized method to estimate the leverage scores. However, the randomized algorithm that was presented in Drineas *et al.* (2012) applies to the $p \ll n$ case, making it challenging to apply these methods to high dimensional settings where p is also very large. Nevertheless this is certainly a promising direction for speeding up ALO-algorithms.

In another line of work, the GCV approach (Craven and Wahba, 1979; Golub *et al.*, 1979) approximates the diagonal elements of $\mathbf{H}$ with $\mathrm{tr}(\mathbf{H})/n$. Computationally efficient randomized estimates of $\mathrm{tr}(\mathbf{H})$ can be produced without having any explicit calculations of this matrix (Deshpande and Girard, 1991; Wahba *et al.*, 1995; Girard, 1998; Lin *et al.*, 2000). The theoretical study of the additional errors that are introduced by these randomized approximations, and the scalable implementations of them, is another promising avenue for future work.

Acknowledgements

We thank the referees for carefully reading the original manuscript and raising important issues which improved the paper significantly. We are also grateful to the laboratory of Edvard Moser, at the Kavli Institute for Systems Neuroscience, at the Norwegian University of Science and Technology in Trondheim, for providing the grid cell data that are presented in Section 5.3.2. KR is grateful to L. Paninski, E. A. Pnevmatikakis and Y. Roudi for fruitful conversations.

KR is supported by National Science Foundation DMS grant 1810888, and the Betty and Marvin Levine Fund.

AM gratefully acknowledges National Science Foundation DMS grant 1810888.

References

Akaike, H. (1974) A new look at the statistical model identification. *IEEE Trans. Autom. Control*, **19**, 716–723.

Alison, W. and Pillow, J. (2017) Capturing the dynamical repertoire of single neurons with generalized linear models. *Neurl Computn*, **29**, 3260–3289.
Allen, D. (1974) The relationship between variable selection and data augmentation and a method for prediction. *Technometrics*, **16**, 125–127.
Bayati, M. and Montanari, A. (2012) The lasso risk for gaussian matrices. *IEEE Trans. Inform. Theory*, **58**, 1997–2017.
Bean, D., Bickel, P. J., El Karoui, N. and Yu, B. (2013) Optimal *m*-estimation in high-dimensional regression. *Proc. Natn. Acad. Sci. USA*, **110**, 14563–14568.
Boucheron, S., Lugosi, G. and Massart, P. (2013) *Concentration Inequalities: a Nonasymptotic Theory of Independence*. New York: Oxford University Press.
Boyd, S. and Vandenberghe, L. (2004) *Convex Optimization*. New York: Oxford University Press.
Breiman, L. and Freedman, D. (1983) How many variables should be entered in a regression equation? *J. Am. Statist. Ass.*, **78**, 131–136.
Brown, E., Nguyen, D., Frank, L., Wilson, M. and Solo, V. (2001) An analysis of neural receptive field plasticity by point process adaptive filtering. *Proc. Natn. Acad. Sci. USA*, **98**, 12261–12266.
Burman, P. (1990) Estimation of generalized additive models. *J. Multiv. Anal.*, **32**, 230–255.
Buzsaki, G. and Moser, E. (2013) Memory, navigation and theta rythm in the hippocampal-entorhinal system. *Nat. Neursci.*, **16**, 130–138.
Cawley, G. and Talbot, N. (2008) Efficient approximate leave-one-out cross-validation for kernel logistic regression. *Mach. Learn.*, **71**, 243–264.
le Cessie, S. and van Houwelingen, J. (1992) Ridge estimators in logistic regression. *Appl. Statist.*, **41**, 191–201.
Craven, P. and Wahba, G. (1979) Estimating the correct degree of smoothing by the method of generalized cross-validation. *Numer. Math.*, **31**, 377–403.
Cunningham, J., Gilja, V., Ryu, S. and Shenoy, K. (2009) Methods for estimating neural firing rates, and their application to brain-machine interface. *Neurl Netwrks*, **22**, 1235–1246.
Cunningham, J., Yu, B., Shenoy, K. and Sahani, M. (2008) Inferring neural firing rates from spike trains using Gaussian processes. In *Advances in Neural Information Processing Systems 20* (eds J. Platt, D. Koller, Y. Singer and S. Roweis). Red Hook: Curran Associates.
Czanner, G., Eden, U., Wirth, S., Yanike, M., Suzuki, W. and Brown, E. (2008) Analysis of between-trial and within-trial neural spiking dynamics. *J. Neurphysiol.*, **99**, 2672–2693.
Deshpande, L. and Girard, D. (1991) Fast computation of cross-validated robust Splines and other non-linear smoothing Splines. In *Curves and Surfaces* (eds P.-J. Laurent, A. Le Méhauté and L. L. Schumaker), pp. 143–148. New York: Academic Press.
DiMatteo, I., Genovese, C. and Kass, R. (2001) Bayesian curve fitting with free-knot splines. *Biometrika*, **88**, 1055–1073.
Dobriban, E. and Wager, S. (2018) High-dimensional asymptotics of prediction: ridge regression and classification. *Ann. Statist.*, **46**, 247–279.
Donoho, D., Maleki, A. and Montanari, A. (2011) Noise sensitivity phase transition. *IEEE Trans. Inform. Theory*, **57**, 6920–6941.
Donoho, D. and Montanari, A. (2016) High dimensional robust m-estimation: asymptotic variance via approximate message passing. *Probab. Theory Reltd Flds*, **166**, 935–969.
Donoho, D. L., Maleki, A. and Montanari, A. (2009) Message passing algorithms for compressed sensing. *Proc. Natn. Acad. Sci. USA*, **106**, 18914–18919.
Drineas, P., Magdon-Ismail, M., Mahoney, M. and Woodruff, D. (2012) Fast approximation of matrix coherence and statistical leverage. *J. Mach. Learn. Res.*, **13**, 3475–3506.
Dunn, B., Morreaunet, M. and Roudi, Y. (2015) Correlations and functional connections in a population of grid cells. *PLOS Computnl Biol.*, **11**, no. 2, article e1004052.
Efron, B. (1983) Estimating the error rate of a prediction rule: improvement on cross-validation. *J. Am. Statist. Ass.*, **78**, 316–331.
Efron, B. (1986) How biased is the apparent error rate of a prediction rule? *J. Am. Statist. Ass.*, **81**, 461–470.
El Karoui, N. (2018) On the impact of predictor geometry on the performance on high-dimensional ridge-regularized generalized robust regression estimators. *Probab. Theory Reltd Flds*, **170**, 95–175.
El Karoui, N., Bean, D., Bickel, P., Lim, C. and Yu, B. (2013) On robust regression with high-dimensional predictors. *Proc. Natn. Acad. Sci. USA*, **110**, 14557–14562.
Frank, I. and Friedman, J. (1993) A statistical view of some chemometric regression tools (with discussion). *Technometrics*, **35**, 109–148.
Friedman, F., Hastie, T. and Tibshirani, R. (2010) Regularization paths for generalized linear models via coordinate descent. *J. Statist. Softwr*, **33**, 1–22.
Gao, Y., Black, M., Bienenstock, E., Shoham, S. and Donoghue, J. (2002) Probabilistic inference of arm motion from neural activity in motor cortex. In *Advances in Neural Information Processing Systems 14* (eds T. G. Dietterich, S. Becker and Z. Ghahramani), pp. 213–220. Cambridge: MIT Press.
Geisser, S. (1975) The predictive sample reuse method with applications. *J. Am. Statist. Ass.*, **70**, 320–328.
Girard, D. (1998) Asymptotic comparison of (partial) cross-validation, GCV and randomized GCV in nonparametric regression. *Ann. Statist.*, **26**, 315–334.

Golub, G., Heath, M. and Wahba, G. (1979) Generalized cross-validation as a method for choosing a good ridge parameter. *Technometrics*, **21**, 215–223.
Gorman, R. and Sejnowski, T. (1988) Analysis of hidden units in a layered network trained to classify sonar targets. *Neurl Netwrks*, **1**, 75–89.
Gu, C. (1992) Cross-validating non-Gaussian data. *J. Computnl Graph. Statist.*, **1**, 169–179.
Gu, C. and Xiang, D. (2001) Cross-validating non-Gaussian data: generalized approximate cross-validation revisited. *J. Computnl Graph. Statist.*, **10**, 581–591.
Hafting, T., Fyhn, M., Molden, S., Moser, M. and Moser, E. (2005) Microstructure of a spatial map in the enthorhinal cortex. *Nature*, **436**, 801–806.
He, L., Qin, W., Xu, P. and Zhou, Y. (2018) alocv: approximate leave-one-out risk estimation. *R Package Version 0.02*.
Hurvich, C. and Tsai, C. (1989) Regression and time series model selection in small samples. *Biometrika*, **76**, 297–307.
Kass, R. E., Ventura, V. and Brown, E. N. (2005) Statistical issues in the analysis of neuronal data. *J. Neurphysiol.*, **94**, 8–25.
Khan, U., Liu, L., Provenzano, F. A., Berman, D., Profacia, C., Sloa, R., Mayeux, R., Duff, K. and Small, S. (2014) Molecular drivers and cortical spread of lateral entorhinal cortex dysfunction in preclinical Alzheimer's disease. *Nat. Neursci.*, **17**, 304–311.
Leeb, H. (2008) Evaluation and selection of models for out-of-sample prediction when the sample size is small relative to the complexity of the data-generating process. *Bernoulli*, **14**, 661–690.
Leeb, H. (2009) Conditional predictive inference post model selection. *Ann. Statist.*, **37**, 2838–2876.
Lin, X., Wahba, G., Xiang, D., Gao, F., Klein, R. and Klein, B. (2000) Smoothing spline ANOVA models for large data sets with Bernoulli observations and the randomized GACV. *Ann. Statist.*, **28**, 1570–1600.
Macke, J., Gerwinn, S., White, L., Kaschube, M. and Bethge, M. (2011) Gaussian process methods for estimating cortical maps. *NeuroImage*, **56**, 570–581.
Maleki, A. (2011) Approximate message passing algorithm for compressed sensing. *PhD Thesis*. Stanford University, Stanford.
Mallows, C. (1973) Some comments on C_p. *Technometrics*, **15**, 661–675.
Meijer, R. and Goeman, J. (2013) Efficient approximate k-fold and leave-one-out cross-validation for ridge regression. *Biometr. J.*, **55**, 141–155.
Moser, E., Kropff, E. and Moser, M. (2008) Place cells, grid cells, and the brain's spatial representation system. *A. Rev. Neursci.*, **31**, 69–89.
Moser, E., Moser, M. B. and Roudi, Y. (2014) Network mechanisms of grid cells. *Phil. Trans. R. Soc.* B, **369**, no. 1635, article 20120511.
Mousavi, A. and Maleki, A. and Baraniuk, R. G. (2018) Consistent parameter estimation for lasso and approximate message passing. *Ann. Statist.*, **46**, 119–148.
Negahban, S., Ravikumar, P., Wainwright, M. and Yu, B. (2012) High-dimensional generalized linear models and the lasso. *Statist. Sci.*, **4**, 538–557.
Nesterov, Y. (2013) *Introductory Lectures on Convex Optimization: a Basic Course*, vol. 87. New York: Springer Science and Business Media.
Nevo, D. and Ritov, Y. (2016) On Bayesian robust regression with diverging number of predictors. *Electron. J. Statist.*, **10**, 3045–3062.
Obuchi, T. and Kabashima, Y. (2016) Cross validation in lasso and its acceleration. *J. Statist. Mech. Theory. Expt.*, **53**, 1–36.
Opper, M. and Winther, O. (2000) Gaussian processes and SVM: mean field results and leave-one-out. In *Advances in Large Margin Classifiers* (eds A. Smola, P. Bartlett, B. Scholkopf and D. Schuurmans), pp. 43–56. Cambridge: MIT Press.
O'Sullivan, F., Yandell, B. and Raynor, W. (1986) Automatic smoothing of regression functions in generalized linear models. *J. Am. Statist. Ass.*, **81**, 96–103.
Paninski, L. (2004) Maximum likelihood estimation of cascade point-process neural encoding models. *Network*, **15**, 243–262.
Paninski, L., Ahmadian, Y., Ferreira, D., Koyama, S., Rahnama Rad, K., Vidne, M., Vogelstein, J. and Wu, W. (2010) A new look at state-space models for neural data. *J. Comput. Neursci.*, **29**, 107–126.
Park, M., Weller, J., Horowitz, G. and Pillow, J. (2014) Bayesian active learning of neural firing rate maps with transformed Gaussian process priors. *Neurl Computn*, **26**, 1519–1541.
Pillow, J. (2007) Likelihood-based approaches to modeling the neural code. *Bayesian Brain: Probabilistic Approaches to Neural Coding*, pp. 53–70. Cambridge: MIT Press.
Pnevmatikakis, E., Rahnama Rad, K., Huggins, J. and Paninski, L. (2014) Fast Kalman filtering and forward-backward smoothing via low-rank perturbative approach. *J. Computnl Graph. Statist.*, **23**, 316–339.
Qian, J., Hastie, T., Friedman, J., Tibshirani, R. and Simon, N. (2013) Glmnet for Matlab. Stanford University, Stanford. (Available from `http://www.stanford.edu/~hastie/glmnet.matlab/`.)
Rahnama Rad, K. and Paninski, L. (2010) Efficient estimation of two-dimensional firing rate surfaces via Gaussian process methods. *Computn Neurl Syst.*, **21**, 142–168.
Rowland, D., Roudi, Y., Moser, M. and Moser, E. (2016) Ten years of grid cells. *A. Rev. Neursci.*, **39**, 19–40.

Schmidt, M., Fung, G. and Rosales, R. (2007) Fast optimization methods for l1 regularization: a comparative study and two new approaches. In *Proc. Eur. Conf. Machine Learning* (eds J. N. Kok, J. Koronacki, R. L. Mantaras, S. Matwin, D. Mladenič and A. Skowron), pp. 286–297. New York: Springer.
Stein, C. (1981) Estimation of the mean of a multivariate normal. *Ann. Statist.*, **9**, 1135–1151.
Stensola, H., Stensola, T., Solstad, T., Froland, K., Moser, M. and Moser, E. (2012) The entorhinal grid map is discretized. *Nature*, **492**, 72–80.
Stone, M. (1974) Cross-validatory choice and assessment of statistical predictions. (with discussion). *J. R. Statist. Soc.* B, **36**, 111–147.
Stone, M. (1977) An asymptotic equivalence of choice of model by cross-validation and Akaike's criterion. *J. R. Statist. Soc.* B, **39**, 44–47.
Su, W., Bogdan, M. and Candes, E. (2017) False discoveries occur early on the Lasso path. *Ann. Statist.*, **45**, 2133–2150.
Tibshirani, R. (1996) Regression shrinkage and selection via the lasso. *J. R. Statist. Soc.* B, **58**, 267–288.
Tibshirani, R. and Taylor, J. (2012) Degrees of freedom in lasso problems. *Ann. Statist.*, **40**, 1198–1232.
Tibshirani, R. J. (2013) The lasso problem and uniqueness. *Electron. J. Statist.*, **7**, 1456–1490.
Van de Geer, S. (2008) High-dimensional generalized linear models and the lasso. *Ann. Statist.*, **2**, 614–645.
Van der Vaart, A. W. (2000) *Asymptotic Statistics*, vol. 3. New York: Cambridge University Press.
Vehtari, A., Gelman, A. and Gabry, J. (2017) High-dimensional generalized linear models and the lasso. *Statist. Comput.*, **5**, 1413–1432.
Vehtari, A., Mononen, T., Tolvanen, V., Sivula, T. and Winther, O. (2016) Bayesian leave-one-out cross-validation approximations for Gaussian latent variable models. *J. Mach. Learn. Res.*, **17**, 3581–3618.
Wahba, G., Johnson, D., Gao, F. and Gong, J. (1995) Adaptive tuning of numerical weather prediction models: randomized GCV in three- and four-dimensional assimilation. *Mnthly Weath. Rev.*, **123**, 3358–3369.
Wainwright, M. (2009) Sharp thresholds for high-dimensional and noisy sparsity recovery using ell_1-constrained quadratic programming (lasso). *IEEE Trans. Inform. Theory*, **55**, 2183–2202.
Weng, H., Maleki, A. and Zheng, L. (2018) Overcoming the limitations of phase transition by higher order analysis of regularization techniques. *Ann. Statist.*, **46**, no. 6A, 3099–3129.
Xiang, D. and Wahba, G. (1996) A generalized approximate cross validation for smoothing splines with non-gaussian data. *Statist. Sin.*, **6**, 675–692.
Zolrowski, D. and Pillow, J. (2018) Scaling the Poisson GLM to massive neural datasets through polynomial approximations. In *Advances in Neural Information Processing Systems 31* (eds S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi and R. Garnett), pp. 3517–3527. Red Hook: Curran Associates.
Zou, H. and Hastie, T. (2005) Regularization and variable selection via the elastic net. *J. R. Statist. Soc.* B, **67**, 301–320.
Zou, H., Hastie, T. and Tibshirani, R. (2007) On the "degrees of freedom" of the lasso. *Ann. Statist.*, **35**, 2173–2192.

Supporting information

Additional 'supporting information' may be found in the on-line version of this article:

'Proofs'.