One-step estimator paths for concave regularization

Matt Taddy

Microsoft Research and the University of Chicago Booth School of Business
faculty.chicagobooth.edu/matt.taddy

The statistics literature of the past 15 years has established many favorable properties for sparse diminishing-bias regularization: techniques which can roughly be understood as providing estimation under penalty functions spanning the range of concavity between ℓ_0 and ℓ_1 norms. However, lasso ℓ_1-regularized estimation remains the standard tool for industrial 'Big Data' applications because of its minimal computational cost and the presence of easy-to-apply rules for penalty selection. In response, this article proposes a simple new algorithm framework that requires no more computation than a lasso path: the path of one-step estimators (POSE) does ℓ_1 penalized regression estimation on a grid of decreasing penalties, but adapts coefficient-specific weights to decrease as a function of the coefficient estimated in the previous path step. This provides sparse diminishing-bias regularization at no extra cost over the fastest lasso algorithms. Moreover, our 'gamma lasso' implementation of POSE is accompanied by a reliable heuristic for the fit degrees of freedom, so that standard information criteria can be applied in penalty selection. We also provide novel results on the distance between weighted-ℓ_1 and ℓ_0 penalized predictors; this allows us to build intuition about POSE and other diminishing-bias regularization schemes. The methods and results are illustrated in extensive simulations and in application of logistic regression to evaluating the performance of hockey players.

1 Introduction

For regression in high-dimensions, it is useful to regularize estimation through a penalty on coefficient size. ℓ_1 regularization (i.e., the lasso of Tibshirani, 1996) is especially popular, with costs that are non-differentiable at their minima and can lead to coefficient solutions of exactly zero. A related approach is concave penalized regularization (e.g. SCAD from Fan and Li 2001 or MCP from Zhang 2010a) with cost functions that are also spiked at zero but flatten for large values (as opposed to the constant increase of an ℓ_1 norm). This yields sparse solutions where large non-zero values are estimated with little bias.

The combination of *sparsity* and *diminishing-bias* is appealing in many settings, and a large literature on concave penalized estimators has developed over the past 15 years. For example, many authors (Fan and Li, 2001; Fan and Peng, 2004) have contributed work on their *oracle properties*, a class of results showing conditions under which coefficient estimates through concave penalization, or in related schemes, will be the same as if you knew the sparse 'truth' (either asymptotically or with high probability). From an information compression perspective, the increased sparsity encouraged by diminishing-bias penalties (since single large coefficients are able to account for the signals of other correlated covariates) leads to lower memory, storage, and communication requirements. Such savings are especially important in distributed computing systems (e.g., Taddy, 2015; Gentzkow et al., 2015); these sorts of Big Data applications provide the original motivation behind our work in this article.

Unfortunately, exact solvers for nonconvex penalized estimation all require significantly more compute time than a standard lasso. This has precluded their use in settings – e.g., text or web-data analysis – where both n (the number of observations) and p (covariate dimension) are very large. As we review in Section 3, recent literature recommends the use of approximate solvers. These approximations take the form of iteratively-weighted-ℓ_1 regularization, where the coefficient-specific weights are based upon results from previous iterations of the approximate solver. Work on one-step estimation (OSE), e.g. by Zou and Li (2008), shows that even a single step of such weighted-ℓ_1 regularization is enough to get solutions that are close to optimal, so long as the pre-estimates are *good enough* starting points. The crux of success is finding starts that are, indeed, good enough.

This article provides a complete framework for sparse diminishing-bias regularization that

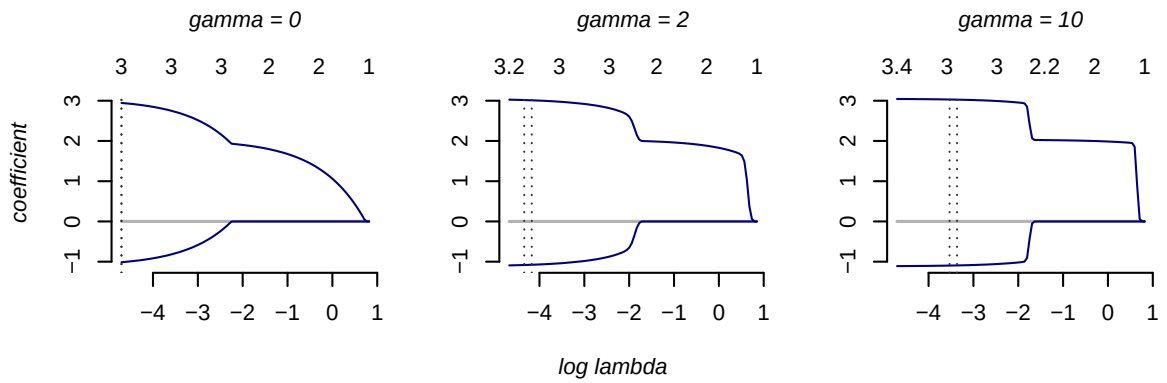

Figure 1: Gamma lasso estimation on $n = 10^3$ observations of $y_i = 4 + 3x_{1i} - x_{2i} + \varepsilon_i$, where $\varepsilon_i \overset{ind}{\sim} \mathrm{N}(0, 1)$ and $\{x_{1i}, x_{2i}, x_{3i}\}$ are marginally standard normal with correlation of 0.9 between covariates (x_{3i} is spurious). The penalty path has $T = 100$ segments, $\lambda^1 = n^{-1} \left|\sum_i x_{1i} y_i\right|$, and $\lambda^{100} = 0.01\lambda^1$. Degrees of freedom are on top and vertical lines mark AICc and BIC selected models (see Section 4).

combines ideas from OSE with the concept of a *regularization path* – a general technique, most famously associated with the LARS algorithm (Efron et al., 2004), that estimates a sequence of models under decreasing amounts of regularization. So long as the estimates do not change too quickly along the path, such algorithms can be very fast to run and are an efficient way to obtain a high-quality *set* of models to choose amongst.

A path of one-step estimators (POSE; Algorithm 1) provides ℓ_1 penalized regression on a grid of decreasing penalties, but adapts coefficient-specific weights to decrease as a function of the coefficient estimated in the previous path step. POSE takes advantage of a natural match between path algorithms and one-step estimation: OSE relies upon inputs being close to the optimal solution, which is precisely the setting where path algorithms are most efficient. We formalize 'close' with a result in Theorem 3.1 that relates weighted-ℓ_1 to ℓ_0 regularization.

This framework allows us to provide

- a *path* of coefficient fits, each element of which corresponds to sparse diminishing-bias regularization estimation under a different level of penalization; where
- obtaining the path of coefficient fits requires *no more computation* than a state-of-the-art ℓ_1 regularization path algorithm; and
- there are good closed-form rules for selection of the optimal penalty level along this path.

The last capability here is derived from a Bayesian interpretation for our *gamma lasso* implementation of POSE, from which we are able to construct information criteria for penalty

selection. We view such tools as an essential ingredient for practical applicability in large-scale industrial machine learning where, e.g., cross-validation is not always viable or advisable.

The remainder of this paper is outlined as follows. Section 2 presents the general regularized regression problem and introduces POSE, our path of one-step estimators algorithm, and the gamma lasso (GL), our implemented version of POSE. Section 3 gives an overview on the relationship between concave and weighted-ℓ_1 regularization. Section 4 provides a Bayesian model interpretation for the gamma lasso, and derives from this model a set of information criteria that can be applied in penalty selection along the regularization path. Finally, we present two empirical studies: an extensive simulation experiment in Section 5, and in Section 6 we investigate the data analysis question: given all goals in the past decade of NHL hockey, what can we say about individual player contributions?

2 Paths of one-step estimators

Denote n response observations as $\mathbf{y} = [y_1, \ldots, y_n]'$ and the associated matrix of p covariates as $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]'$, with rows $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]'$ and columns $\boldsymbol{\chi}_j = [x_{1j}, \ldots, x_{nj}]'$. Since the size of penalized β_j depends upon the units of x_{ij}, it is common to scale the coefficient by $\mathrm{sd}(\boldsymbol{\chi}_j)$, the standard deviation of the j^{th} column of $\mathbf{X}$, before assessing its penalty cost. We ignore this for notational convenience, but if desired you can simply replace x_{ij} by $x_{ij}/\mathrm{sd}(\boldsymbol{\chi}_j)$ throughout. Write $\eta_i = \alpha + \mathbf{x}_i'\boldsymbol{\beta}$ as the linear model for observation i. Denote with $l(\alpha, \boldsymbol{\beta})$, or shortened to $l(\boldsymbol{\eta})$, an objective function proportional to the deviance. For example, in Gaussian (linear) regression, $l(\boldsymbol{\eta})$ is the sum-of-squares $0.5\sum_i (y_i - \eta_i)^2$ and in binomial (logistic) regression, $l(\boldsymbol{\eta}) = -\sum_i [\eta_i y_i - \log(1 + e^{\eta_i})]$ for $y_i \in [0, 1]$.

A penalized estimator is the solution to

$$\underset{\alpha,\beta_j \in \mathbb{R}}{\operatorname{argmin}} \quad \left\{ l(\alpha, \boldsymbol{\beta}) + n\lambda \sum_{j=1}^{p} c(\beta_j) \right\}, \tag{1}$$

where $\lambda > 0$ controls overall penalty magnitude and $c(\cdot)$ is the coefficient cost function.

A few common cost functions are: ℓ_2, $c(\beta) \propto \beta^2$ (ridge, Hoerl and Kennard, 1970); ℓ_1, $c(\beta) \propto |\beta|$ (lasso, Tibshirani, 1996); the 'elastic net' mixture of ℓ_1 and ℓ_2 (Zou and Hastie,

2005); and the log penalty $c(\beta) \propto \log(1+\gamma|\beta|)$ (Candes et al., 2008). Those that have a non-differentiable spike at zero (all but ridge) lead to sparse estimators, with some coefficients set to exactly zero. The curvature of the penalty away from zero dictates then the weight of shrinkage imposed on the nonzero coefficients: ℓ_2 costs increase with coefficient size, lasso's ℓ_1 penalty has zero curvature and imposes constant shrinkage, and as curvature goes towards $-\infty$ one approaches the ℓ_0 penalty of subset selection. In this article we are primarily interested in *concave cost functions*, like the log penalty, occupying the range between ℓ_1 and ℓ_0 penalties.

Penalty size, λ, acts as a *squelch*: it suppresses noise to focus on the true input signal. Large λ lead to very simple model estimates, while as $\lambda \to 0$ we approach maximum likelihood estimation (MLE). Since you don't know optimal λ, practical application of penalized estimation requires a *regularization path*: a $p \times T$ field of $\hat{\boldsymbol{\beta}}$ estimates, say $\hat{\boldsymbol{\beta}}|_\lambda$, obtained while moving from high to low penalization along $\lambda^1 > \lambda^2 \ldots > \lambda^T$. These paths begin at λ^1 set to infimum λ such that (1) is minimized at $\hat{\boldsymbol{\beta}}|_\lambda = \mathbf{0}$, and move to a pre-specified λ^T (e.g., $\lambda^T = 0.01\lambda^1$).

Our path of one-step estimators (POSE) framework is in Algorithm 1.

Algorithm 1 POSE

Initialize $\lambda^1 = \inf\left\{\lambda : \ \hat{\boldsymbol{\beta}}|_\lambda = \mathbf{0}\right\}$, so that $\hat{\boldsymbol{\beta}}_1 = \mathbf{0}$. Say $\hat{S}_t = \{j : \hat{\beta}_j^t \neq 0\}$

Set step size $0 < \delta < 1$.

for $t = 2 \ldots T$:

$$\lambda^t = \delta\lambda^{t-1}$$

$$\omega_j^t = \begin{cases} c'(|\hat{\beta}_j^{t-1}|) \text{ for } j \in \hat{S}_{t-1} \\ 1 \text{ for } j \in \hat{S}_{t-1}^c \end{cases} \tag{2}$$

$$\left[\hat{\alpha}, \hat{\boldsymbol{\beta}}\right]^t = \underset{\alpha,\beta_j \in \mathbb{R}}{\operatorname{argmin}} \ \ l(\alpha, \boldsymbol{\beta}) + n\sum_j \lambda^t \omega_j^t |\beta_j| \tag{3}$$

We are assuming a cost function that is differentiable away from zero and has been scaled such that $\lim_{b\to 0} c'(b) = 1$. Since POSE starts at simple ℓ_1 penalization (i.e., with $c(\beta_j) = 1$), our initial penalty weight is available analytically as $\lambda^1 = n^{-1}\max\left\{\left|\partial l(\boldsymbol{\beta})/\partial\beta_j|_{\mathbf{0}}\right|, \ j = 1 \ldots p\right\}$, the maximum mean absolute gradient evaluated at $\boldsymbol{\beta} = \mathbf{0}$; see the supplement for more detail.

Section 3 details how POSE relates to concave regularization. For some quick intuition, consider POSE with a concave cost function (e.g., the log penalty in Figure 2). The derivative

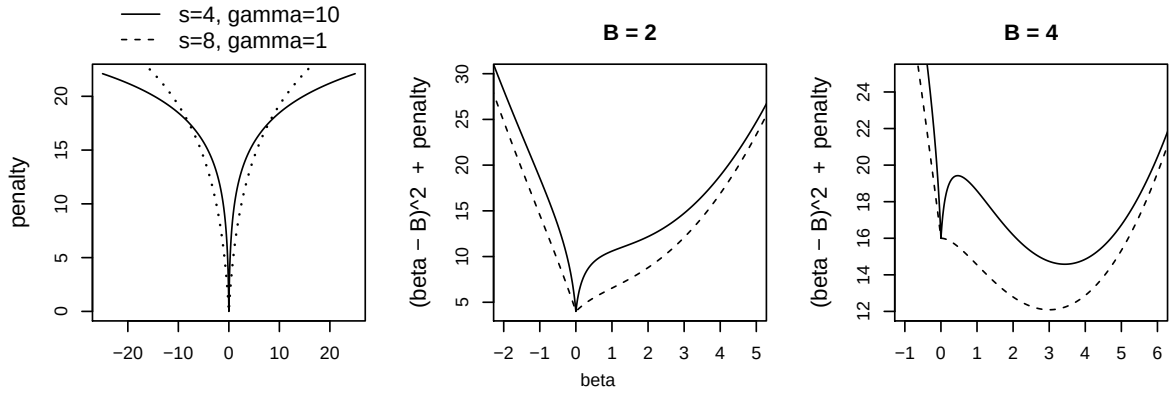

Figure 2: Log penalties $c(\beta) = s\log(1+\gamma|\beta|)$ and penalized objectives $(\beta - B)^2 + c(\beta)$.

$c'(|\hat{\beta}|)$ will be positive but decreasing with larger values of $\hat{\beta}$, such that the *weight* on the ℓ_1 penalty for $\hat{\beta}_j^t$ will *diminish* with the size of $|\hat{\beta}_j^t|$. This implies that coefficient estimates later in the path will be less biased towards zero if that coefficient has a large value earlier in the path.

2.1 The gamma lasso

The gamma lasso (GL) specification for POSE is based upon the log penalty,

$$c(\beta_j) = \gamma^{-1}\log(1+\gamma|\beta_j|), \tag{4}$$

where $\gamma > 0$. This appears under a variety of parameterizations and names in the literature; see Mazumder et al. (2011) and applications in Friedman (2008), Candes et al. (2008), Cevher (2009), Taddy (2013) and Armagan et al. (2013). The penalty is nondifferentiable at zero and concave away from zero with curvature $c''(|b|) = -\gamma/(1+\gamma|b|)^2$ and gradient $c'(|b|) = 1/(1+\gamma|b|)$ (note that $\lim_{b\to 0} c'(b) = 1$ as required). This spans the range from ℓ_1 to ℓ_0 penalization: $\lim_{\gamma\to 0} c(b) = |b|$, while large γ yield ℓ_0-type costs with $c'(|b|) \approx 0\ \forall b \neq 0$. (Note that $\lim_{\gamma\to\infty} c(b) = 0$; however, POSE yields forward stepwise selection in this limit).

GL simply replaces line (2) in Algorithm 1 with

$$\omega_j^t = \left(1+\gamma|\hat{\beta}_j^{t-1}|\right)^{-1} \quad j = 1\dots p \tag{5}$$

Behavior of the resulting paths is governed by γ, which we refer to as the penalty *scale*. Under $\gamma = 0$, GL is just the usual lasso. Bias diminishes faster for larger γ. At the extreme, $\gamma = \infty$

yields a greedy subset selection routine where a coefficient is unpenalized in all segments after it first becomes nonzero. Figure 1 shows solutions in a simple problem.

Each gamma lasso path segment is solved through coordinate descent (see supplement). The algorithm is implemented in `c` as part of the `gamlr` package for R. Usage of `gamlr` mirrors that of its convex penalty analogue `glmnet` (Friedman et al., 2010), the fantastic and widely used package for costs between ℓ_1 and ℓ_2 norms. In the lasso case ($\gamma = 0$), the two algorithms are essentially equivalent.

3 Weighted-ℓ_1 approximations to concave penalization

Concave penalties such as the log penalty, which have a gradient that is decreasing with absolute coefficient size, yield the 'diminishing-bias' property discussed above. It is *the* reason why one would use concave penalization instead of ℓ_1 or convex alternatives.

Unfortunately, such penalties can overwhelm the convex likelihood and produce a nonconvex minimization objective; see Figure 2. This makes computation difficult. For example, one run of SCAD via the `ncvreg` R package (Breheny and Huang, 2011) for the simulation in Section 5 requires around 10 minutes, compared to less than a second for lasso (or gamma lasso). The most efficient exact solver that we've found is the `sparsenet` of Mazumder et al. (2011), also implemented in R, which first fits a lasso path and, for each segment on this path, adapts coefficient estimates along a second path of increasing penalty concavity. However, `sparsenet` relies upon sequential solution over a large set of specifications and its compute cost remains much higher than for the [gamma] lasso.

Local linear approximation (LLA; e.g., Candes et al., 2008) algorithms replace the nonconvex cost function c with its tangent at the current estimate, $c'(\hat{\beta}_j)\beta_j$. The objective is then just a weighted ℓ_1 penalized loss. An exact LLA solver iterates between updating $c'(\hat{\beta})$ and solving the implied ℓ_1 penalized minimization problem. Zou and Li (2008) present numerical and theoretical evidence that LLA can provide near-optimal solutions even if you *stop it after one iteration*. This is an example of one-step estimation (OSE; Bickel, 1975), wherein you take as your estimator the first step of an iterative approximation to some objective. Early-stopping can be as good as the full solution *if* the initial estimates are good enough.

OSE and similar ideas have had a resurgence in the concave penalization literature recently, motivated by the need for faster estimation algorithms. Fan et al. (2014) consider early-stopping of LLA for folded concave penalization and show that, under strong sparsity assumptions about true $\boldsymbol{\beta}$ and given appropriate initial values, OSE LLA is with high probability an oracle estimator. Zhang (2010,2013) investigates 'convex relaxation' iterations, where estimates under convex regularization are the basis for weights in a subsequent penalized objective. Wang et al. (2013) propose a two step algorithm that feeds lasso coefficients into a linear approximation to folded concave penalization. These OSE methods are all closely related to the adaptive lasso (AL; Zou, 2006), which does weighted-ℓ_1 minimization under weights $\omega_j = 1/|\hat{\beta}_j^0|$, where $\hat{\beta}_j^0$ is an initial guess at the coefficient value. The original AL paper advocates using MLE estimates for initial values, while Huang et al. (2008) suggest using marginal correlations $\hat{\beta}_j^0 = \mathrm{cor}(\mathbf{x}_j, \mathbf{y})$; this marginal AL algorithm is included in our simulations of Section 5.

The main point is that OSE LLA, or a two-step estimator starting from $\hat{\boldsymbol{\beta}} = 0$, or any version of the adaptive lasso, are all interpretable as weighted-ℓ_1 penalization with weights equal to something like $c'(\beta^0)$ for initial coefficient guess β^0. The algorithms proposed in Section 2, POSE and GL, take advantage of an available source of initial values in any path estimation algorithm – the solved values from the previous path iteration. Our simulations in Section 5 show that this efficient strategy works as well or better than expensive exact solvers. In the next section, we provide some theoretical intuition on why it works.

3.1 Comparison between weighted-ℓ_1 and ℓ_0 penalization

Our oracle comparator is estimation under ℓ_0 costs, $c(\beta_j) = \mathbb{1}_{\{\beta_j \neq 0\}}$, for which global solution is impractical. We treat the design $\mathbf{X}$ as *fixed*, and make no assumptions about the distribution for $\mathbf{y}|\mathbf{X}$ (not even independence between observations; see remarks). The question we address is thus more operational than statistical: for a fixed sample, what is the distance between an easy-to-find weighted-ℓ_1 penalized solution and the infeasible ℓ_0-penalized optimum?

In the theoretical setups more familiar to statisticians, one bounds the expected difference either between fitted coefficients and some assumed 'true' model parameters or between model predictions and future response. Both of these evaluations are important, and they have been studied extensively elsewhere. The text by Bühlmann and van de Geer (2011) includes a com-

prehensive treatment of such prediction and estimation risk for ℓ_1 and weighted-ℓ_1 penalized linear models, and we refer the interested reader there for this material and abundant references to other relevant work. This literature usually assumes a truth that is (at least approximately) linear and sparse in the available covariates. The assumption of true sparsity is dubious in many realistic applications, but it is necessary inasmuch as, with finite data, most parameters in a high-dimensional model cannot be reliably measured as different from zero.

Instead, we are mostly interested in weighted-ℓ_1 penalization as a way to obtain fits that are as sparse as possible without compromising prediction, regardless of whether the data generating process is sparse. By comparing to an ideal optimization objective rather than to some true model, we are able to present a finite-sample fully-nonparametric result with a straightforward intuitive proof. Obviously, this result is useful only if an ℓ_0 penalized estimator would work well for the problem at hand. However, there is theoretical support for optimality of ℓ_0 penalization in a variety of high-dimensional prediction settings (e.g, Mallows, 1973; Efron, 2004). Indeed, our simulation studies show that ℓ_0 oracles are nearly always the best performing option in terms of both prediction and estimation error. In the case where an ℓ_0 oracle does poorly, we would usually argue against penalized linear models as a strategy anyways.

3.1.1 Approximation to an ℓ_0 oracle

For $S \subset \{1 \dots p\}$ with cardinality $|S| = s$ and complement $S^c = \{1 \dots p\} \setminus S$, denote vectors restricted to covariates in S as $\beta_S = [\beta_j : j \in S]'$, matrices as $\mathbf{X}_S$, etc. Use $\boldsymbol{\beta}^S$ to denote the coefficients for ordinary least-squares (OLS) restricted to S: that is, $\boldsymbol{\beta}^S_S = (\mathbf{X}'_S\mathbf{X}_S)^{-1}\mathbf{X}'_S\mathbf{y}$ and $\beta^S_j = 0 \; \forall \; j \notin S$. Moreover, $\mathbf{e}^S = \mathbf{y} - \mathbf{X}\boldsymbol{\beta}^S = (\mathbf{I} - \mathbf{H}^S)\mathbf{y}$ are residuals and $\mathbf{H}^S = \mathbf{X}_S(\mathbf{X}'_S\mathbf{X}_S)^{-1}\mathbf{X}'_S$ the projection matrix from OLS on S. Use $|\cdot|$ and $\|\cdot\|$ for ℓ_1 and ℓ_2 norms.

We use the following result for iterative *stagewise* regression; proof is in the supplement.

LEMMA 3.1. *Say* $\mathrm{MSE}_S = \|\mathbf{X}\boldsymbol{\beta}^S - \mathbf{y}\|^2/n$ *and* $\mathrm{cov}(\boldsymbol{\chi}_j, \mathbf{e}^S) = \boldsymbol{\chi}'_j(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^S)/n$ *are sample variance and covariances. Then for any* $j \in 1 \dots p$*,*

$$\mathrm{cov}^2(\boldsymbol{\chi}_j, \mathbf{e}^S) \leq \mathrm{MSE}_S - \mathrm{MSE}_{S \cup j}$$

In addition, we need to define *restricted eigenvalues* (RE) on the gram matrix $\mathbf{X}'\mathbf{X}/n$. This

RE matches the 'adaptive restricted eigenvalues' of Bühlmann and van de Geer (2011). Similar quantities are common in the theory of regularized estimators; Raskutti et al. (2010) show that similar conditions hold given $\omega_{S_c}^{\min} = 1$ with high probability for $\mathbf{X}$ drawn from a broad class of Gaussian distributions. Bickel et al. (2009) provide a nice overview of sufficient conditions, and Bühlmann and van de Geer (2011) have extensive discussion and examples.

DEFINITION 3.1. *The restricted eigenvalue is* $\phi^2(L,S) = \min_{\{\mathbf{v}:\mathbf{v}\neq\mathbf{0},\ |\mathbf{v}_{S^c}|\leq L\sqrt{s}\|\mathbf{v}_S\|\}} \frac{\|\mathbf{X}\mathbf{v}\|^2}{n\|\mathbf{v}\|^2}$.

Finally, we bound the distance between prediction from ℓ_0 and weighted-ℓ_1 regularization.

THEOREM 3.1. *Consider squared-error loss* $l(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2$*, and suppose* $\boldsymbol{\beta}^S$ *minimizes the* ℓ_0 *penalized objective* $l(\boldsymbol{\beta}) + n\nu\sum_{j=1}^p \mathbb{1}_{\{\beta_j\neq 0\}}$ *with* $|S| = s < n$*. Write* $\hat{\boldsymbol{\beta}}$ *as solution to the weighted-*ℓ_1 *minimization* $l(\boldsymbol{\beta}) + n\lambda\sum_j \omega_j|\beta_j|$.

Then $\omega_{S^c}^{\min}\lambda > \sqrt{2\nu}$ *while* $\phi^2(L,S) > 0$*, with* $L = \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}}\left(\omega_{S^c}^{\min} - \sqrt{2\nu}/\lambda\right)^{-1}$*, implies*

$$\frac{\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)\|^2}{n} \leq \frac{4\lambda^2\|\boldsymbol{\omega}_S\|^2}{\phi^2(L,S)}. \tag{6}$$

Proof. From the definitions of $\hat{\boldsymbol{\beta}}$ and $\boldsymbol{\beta}^S$,

$$\begin{aligned}\frac{1}{2}\|\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{y}\|^2 + n\lambda\sum_j \omega_j|\hat{\beta}_j| &= \frac{\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)\|^2}{2} + \frac{\|\mathbf{e}^S\|^2}{2} - \hat{\mathbf{y}}'\mathbf{e}^S + n\lambda\sum_j \omega_j|\hat{\beta}_j| \\ &\leq \frac{1}{2}\|\mathbf{e}^S\|^2 + n\lambda\sum_{j\in S}\omega_j|\beta_j^S|\end{aligned} \tag{7}$$

Since $\hat{\mathbf{y}}'\mathbf{e}^S = \hat{\mathbf{y}}'(\mathbf{I} - \mathbf{H}^S)\mathbf{y} = \hat{\boldsymbol{\beta}}'\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^S) = \sum_{j\in S^c}\hat{\beta}_j\boldsymbol{\chi}_j'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^S)$, we can apply Lemma 3.1 followed by $\boldsymbol{\beta}^S$ being optimal under ℓ_0 penalty ν to get

$$\left(\frac{\boldsymbol{\chi}_j'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^S)}{n}\right)^2 \leq \mathrm{MSE}_S - \mathrm{MSE}_{S\cup j} < 2\nu \quad \forall\, j \tag{8}$$

so that $|\hat{\mathbf{y}}'\mathbf{e}^S| = |\hat{\boldsymbol{\beta}}_{S^c}\mathbf{X}_{S^c}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^S)| < n\sqrt{2\nu}|\hat{\boldsymbol{\beta}}_{S^c}|$. Applying this inside (7),

$$\frac{1}{2}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)\|^2 + n\left(\omega_{S^c}^{\min}\lambda - \sqrt{2\nu}\right)|\hat{\boldsymbol{\beta}}_{S^c}| \ \leq\ n\lambda\sum_{j\in S}\omega_j|\hat{\beta}_j - \beta_j^S| \ \leq\ n\lambda\|\boldsymbol{\omega}_S\|\|\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^S\|. \tag{9}$$

Given $\omega_{S^c}^{\min}\lambda > \sqrt{2\nu}$, difference $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S$ is in the RE support for $L = \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}}(\omega_{S^c}^{\min} - \sqrt{2\nu}/\lambda)^{-1}$ and thus $\|\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^S\| \leq \|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)/\sqrt{n}\|/\phi(L,S)$. Finally, applying this inside (9) yields

$$\frac{1}{2}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)\|^2 \;\leq\; \frac{\sqrt{n}\lambda\|\boldsymbol{\omega}_S\|\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)\|}{\phi(L,S)}. \tag{10}$$

Dividing each side by $\sqrt{n}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)\|/2$ and squaring gives the result. □

Remarks

- Theorem 3.1 is finite sample exact. Distinguishing it from related results in the literature, it is also completely *non-parametric* – it makes no reference to the true distribution of $\mathbf{y}|\mathbf{X}$. Indeed, if we make such assumptions, Theorem 3.1 provides bounds on the distance between a weighted-lasso and optimal prediction. The next remark is an example.

- Assume that $\mathbf{y} \sim (\boldsymbol{\eta}, \sigma^2\mathbf{I})$ – y_i independent with mean μ_i and shared variance σ^2. The C_p formula of Mallows (1973) gives $\mathrm{MSE}_S + 2s\sigma^2/n$ as an unbiased estimate of residual variance. Following Efron (2004), this implies $\nu = \sigma^2/n$ is the optimal ℓ_0 penalty for minimizing prediction error. Theorem (3.1) applies directly, with $L = \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}}\left(\omega_{S^c}^{\min} - \sqrt{2}\sigma/(\lambda\sqrt{n})\right)^{-1}$, to give a bound on the distance between weighted-ℓ_1 estimation and C_p-optimal prediction. Note that, since the condition on minimum S^c weights has become $\omega_{S^c}^{\min} > (\sigma/\lambda)\sqrt{2/n}$, comparison to C_p suggests we can use larger γ with large n or small σ.

- Plausibility of the restricted eigenvalue assumption $\phi(L,S) > 0$ depends upon L. It is less restrictive if we can reduce $\|\boldsymbol{\omega}_S\|$ without making $\omega_{S_c}^{\min}$ small. *This is a key motivation for the POSE algorithm:* if covariates with nonzero $\hat{\beta}_j$ for large λ (i.e., early in the path) can be assumed to live in S, then increasing $\gamma > 0$ will improve prediction. Of course, the moment that λ becomes small enough that elements of S_c get nonzero $\hat{\beta}$, then larger γ can lead to overfit. For this reason it is essential that we have tools for choosing optimal λ. In the following sections, we describe both cross-validation and information criteria for penalty selection.

- For the lasso, $\boldsymbol{\omega} = \mathbf{1}$ and $\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^S)\|^2/n \leq 4\lambda^2 s/\phi^2(L,S)$ with $L = (1 - \sqrt{2\nu})^{-1}$. This bound depends only upon s, but forcing $\phi^2(L,S) > 0$ becomes more restrictive for larger p.

- There is no notion of a 'true' model here and this result has nothing to say about the estimation error on individual parameters. We again refer the reader to Bühlmann and van de

Geer (2011), and note that for minimizing estimation error they recommend ℓ_1 weights derived from lasso fits at smaller λ (i.e., the opposite of GL and POSE, where weights are derived from fits at slightly larger λ). However, we consider estimation error in our simulation study and find that the GL algorithms do well compared to MCP and the adaptive lasso. Moreover, in the supplemental material we adapt standard results from Wainwright (2006, 2009) to show how reducing $\|\boldsymbol{\omega}_S\|$ without making $\omega_{S_c}^{\min}$ small can lead to lower false discovery rates with respect to the ℓ_0 oracle. Unfortunately, this relies upon fairly strict design restrictions.

4 Penalty selection

Lasso, the gamma lasso, and related sparse regularization estimators do not actually *do* model selection; rather, they can be used to obtain paths of estimates corresponding to different levels of penalization. Each penalty level corresponds to a different 'model' and we must select the optimal choice from these candidates.

K-fold cross-validation (CV; e.g., Efron, 2004) is the most common technique for penalty selection, and it does a good job. However, there are many scenarios where we might want an analytic alternative to CV. For example, if a single fit is expensive then doing it K times will be impractical. More subtly, truly Big Data are distributed. Algorithms can be designed to work in parallel on subsets (e.g., Taddy, 2015) but a bottleneck results if you need to communicate across machines for CV experimentation. Finally, CV can lead to over-fit for unstable algorithms whose results change dramatically in response to data jitter; see Breiman (1996) for a classic discussion and the supplement for an overview.

An important feature of the standard ℓ_1 lasso is that it comes with a simple approximation for the estimation degrees-of-freedom (*df*) at any λ: the number of nonzero estimated coefficients (see Zou et al., 2007). These *df* can combined with the fitted deviance to create information criteria, such as the AIC or BIC, that provide alternative tools for penalty selection.

This section derives the gamma lasso as approximately maximizing the posterior for a hierarchical Bayesian model, and uses this interpretation to obtain a heuristic degrees-of-freedom for each estimate along the GL path. These GL *df* can be input to information criteria for model selection. In particular, our extensive simulations demonstrate that the GL *df* can be used with

the corrected AICc of Hurvich and Tsai (1989) to obtain out-of-sample predictive performance that is as good or better than that from cross-validated predictors.

4.1 Bayesian model interpretation

Consider a model where each β_j is assigned a Laplace distribution prior with scale $\tau_j > 0$,

$$\beta_j \sim \mathrm{La}\left(\tau_j\right) = \frac{\tau_j}{2} \exp\left[-\tau_j|\beta_j|\,\right]. \tag{11}$$

Typically, scale parameters $\tau_1 = \ldots = \tau_p$ are set as a single value, say $n\lambda/\phi$ where ϕ is the dispersion (e.g. Gaussian variance σ^2 or 1 for the binomial). Posterior maximization under the prior in (11) is ℓ_1 regularized estimation (e.g., Park and Casella, 2008).

Instead of a single shared scale, assume an independent gamma $\mathrm{Ga}(s, 1/\gamma)$ hyperprior with 'shape' s and 'scale' γ for each τ_j, such that $\mathbb{E}[\tau_j] = s\gamma$ and $\mathrm{var}(\tau_j) = s\gamma^2$. The *joint* coefficient-scale prior is

$$\pi(\beta_j, \tau_j) = \mathrm{La}\left(\beta_j;\ \tau_j\right) \mathrm{Ga}\left(\tau_j;\ s, \gamma^{-1}\right) = \frac{1}{2\Gamma(s)} \left(\frac{\tau_j}{\gamma}\right)^s \exp\left[-\tau_j(\gamma^{-1} + |\beta_j|)\right]. \tag{12}$$

The gamma hyperprior is conjugate here, implying a $\mathrm{Ga}\left(s+1,\ 1/\gamma + |\beta_j|\right)$ posterior for $\tau_j \mid \beta_j$ with conditional posterior mode (MAP) at $\hat{\tau}_j = \gamma s/(1+\gamma|\beta_j|)$.

Consider joint MAP estimation of $[\boldsymbol{\tau}, \boldsymbol{\beta}]$ under the prior in (12), where we've suppressed α for simplicity. By taking negative logs and removing constants, this is equivalent to solving

$$\underset{\beta_j \in \mathbb{R},\ \tau_j \in \mathbb{R}^+}{\operatorname{argmin}} \frac{l(\boldsymbol{\beta})}{\phi} + \sum_j \left[\tau_j(\gamma^{-1} + |\beta_j|) - s\log(\tau_j)\right]. \tag{13}$$

It is straightforward to show (supplement) that the $\boldsymbol{\beta}$ which solves (13) is also the solutions to the log-penalized objective

$$\underset{\beta_j \in \mathbb{R}}{\operatorname{argmin}} \ \ \phi^{-1} l(\boldsymbol{\beta}) + \sum_j s\log(1+\gamma|\beta_j|), \tag{14}$$

such that the log penalty is interpretable as a *profile* MAP estimate.

4.2 Degrees of freedom

For prediction rules, say $\hat{y}_i$, that are suitably stable (i.e., Lipschitz; see Zou et al., 2007), the SURE framework of Stein (1981) applies and $df = \mathbb{E}\left[\sum_i \partial\hat{y}_i/\partial y_i\right]$. Consider a single coefficient β estimated via least-squares under ℓ_1 penalty τ. Write gradient at zero $g = -\sum_i x_i y_i$ and curvature $h = \sum_i x_i^2$ and set $\varsigma = -\texttt{sign}(g)$. The prediction rule is $\hat{y} = x(\varsigma/h)(|g| - \tau)_+$ with derivative $\partial\hat{y}_i/\partial y = x_i^2/h\mathbb{1}_{[|g|>\tau]}$, so that the SURE expression yields $df = \mathbb{E}\left[\mathbb{1}_{[|g|>\tau]}\right]$. This expectation is taken with respect to the *unknown true* distribution over $\mathbf{y}|\mathbf{X}$, not that estimated from the observed sample. However, one can evaluate this expression at observed gradients as an unbiased estimator for the true *df* (e.g., Zou et al., 2007).

This motivates our heuristic *df* in weighted-ℓ_1 regularization: the *prior* expectation for the number ℓ_1 penalty dimensions, $\tau_j = \lambda\omega_j$, that are less than their corresponding absolute gradient dimension. Referring to our Bayesian model above, each τ_j is iid $\mathrm{Ga}(s, 1/\gamma)$ in the prior, leading to the GL degrees of freedom

$$df^t = \sum_j \mathrm{Ga}(|g_j|;\ n\lambda^t/(\gamma\phi), 1/\gamma), \tag{15}$$

where $\mathrm{Ga}(\ \cdot\ ;\ \mathrm{shape}, 1/\mathrm{scale})$ is the Gamma cumulative distribution function and g_j is an estimate of the j^{th} coefficient gradient evaluated at $\hat{\beta}_j = 0$. Note that the number of unpenalized variables (e.g., 1 for α) should also be added to the total estimation df. For orthogonal covariates, g_j is just the marginal gradient at zero. In the non-orthogonal case, where $g_j = g_j(0)$ becomes a function of all of the elements of $\hat{\boldsymbol{\beta}}$, we plug in the most recent g_j at which $\hat{\beta}_j^t = 0$: this requires no extra computation and has the advantage of maintaining $df^t = \hat{p}^t$ for $\gamma = 0$.

4.3 Selection via information criteria

An information criterion is an attempt to approximate divergence between the unknown true data generating process and our parametric approximation; see the supplement for an overview. These take the form

$$l(\hat{\boldsymbol{\beta}}) + k(df) \tag{16}$$

where k is the cost on the degrees-of-freedom associated with $\hat{\boldsymbol{\beta}}$ and l is the negative log likelihood. The AIC of Akaike (1973) uses $k(df) = df$ while the BIC of Schwarz (1978) uses $k(df) = \log(n)df/2$. As detailed in Flynn et al. (2013), the corrected AICc with $k(df) = df \times n/(n - df - 1)$ does a better job than the AIC or BIC in choosing the optimal model for prediction when df is large. Alternatively, the BIC is often preferred for accurate support recovery or avoiding false discovery; see, e.g., Zou et al. (2007).

5 Simulation experiment

We consider continuous-response data simulated from a $p = 1000$ dimensional linear model

$$y \sim \mathrm{N}\left(\mathbf{x}'\boldsymbol{\beta}, \sigma^2\right) \text{ where } \beta_j = (-1)^j \exp\left(-\frac{j}{\kappa}\right) \mathbb{1}_{[j \leq J]} \text{ for } j = 1 \ldots p \ . \tag{17}$$

Each simulation draws n means $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$ and two independent samples $\mathbf{y}, \tilde{\mathbf{y}} \sim \mathrm{N}(\boldsymbol{\eta}, \sigma^2\mathbf{I})$; the first sample is used for training and we evaluate prediction error on the second sample. Our experiment includes all possible combinations of the following configuration options:

- the sample size is $n = 100$ or $n = 1000$;
- the simulation models is either *dense*, with $J = p$ so that all true coefficients are nonzero, or *sparse*, with $J = n/10$ for either 10 or 100 nonzero coefficients;
- defining $\mathbf{z}_i \sim \mathrm{N}(\mathbf{0}, \boldsymbol{\Sigma})$ for $i = 1 \ldots n$, the regression inputs $\mathbf{x}_i$ are generated as either *continuous* $x_{ij} = z_{ij}$ or *binary* $x_{ij} \overset{ind}{\sim} \mathrm{Bern}\left(1/(1 + e^{-z_{ij}})\right)$;
- error variance σ^2 is defined through *signal-to-noise* (*s2n*) ratios $\mathrm{sd}(\boldsymbol{\eta})/\sigma$ of $1/2$, 1, or 2;
- design multicollinearity is parametrized via $\Sigma_{jk} = \rho^{|j-k|}$, with ρ of 0, 0.5, or 0.9;
- the rate of coefficient decay is specified by κ of 10, 50, 100, or 200.

This implies a total of 288 different models, and we simulate and estimate 1000 times for each.

Our simulation models include various levels for both true sparsity – adjusted through the threshold J – and the *effective sparsity* dictated by κ, which controls the rate of coefficient decay. At $\kappa = 10$ only 23 coefficients have absolute value larger than 0.1, while at $\kappa = 200$ there are 460 coefficients larger than this threshold. The strictly dense $J = p$ models have all nonzero true coefficients but it will be useless to estimate many of them when $p = n$ or $p > n$.

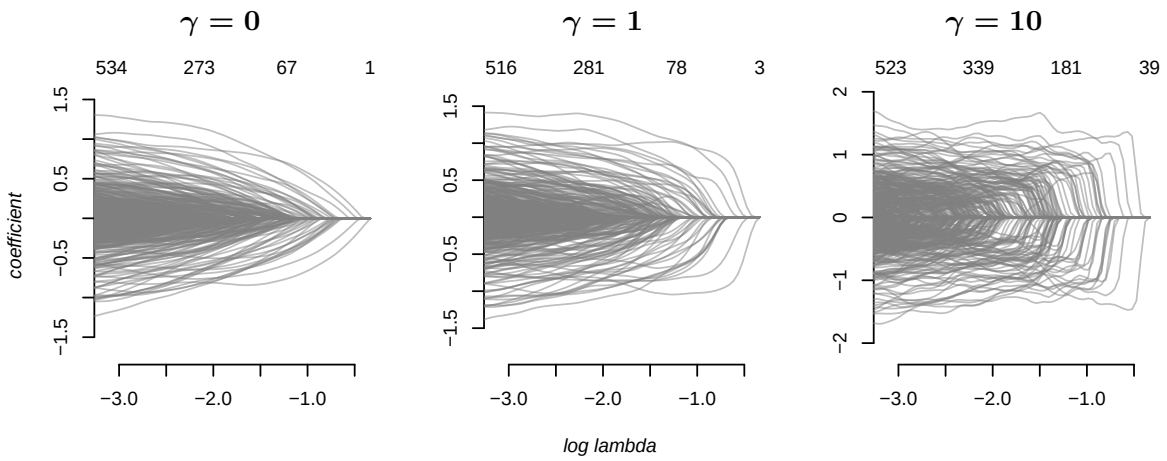

Figure 3: Regularization paths for simulation example. Degrees of freedom df^t are along the top.

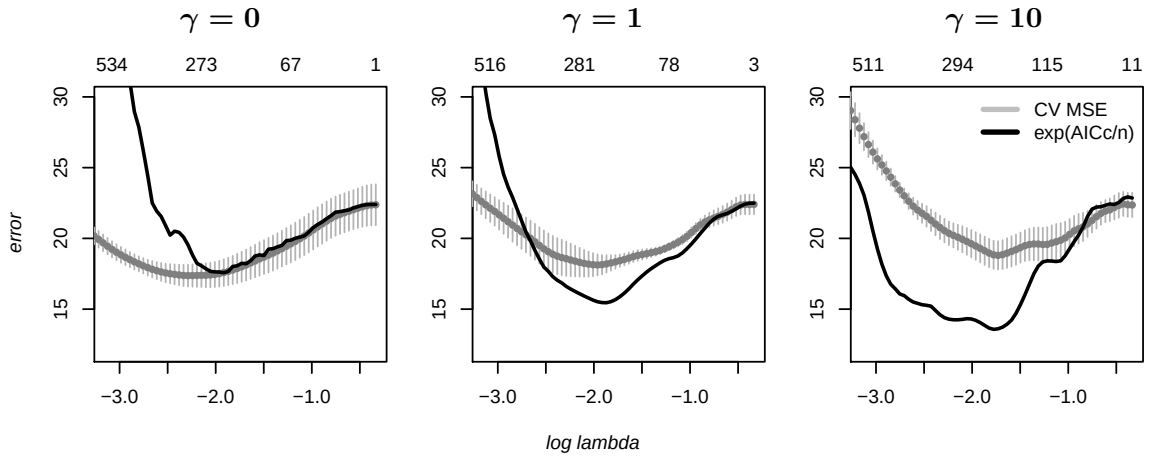

Figure 4: 5-fold CV and AICc for a simulation example. Points-and-bars show mean OOS MSE $\pm$1se.

Figures 3 and 4 illustrate GL paths for a single dataset, generated from our *dense* model with *binary* design, $\mathrm{sd}(\eta)/\sigma = 1$, $\rho = 0.9$, and $\kappa = 50$. In Figure 3, increasing γ leads to 'larger shouldered' paths where estimates move quickly to MLE for the nonzero-coefficients. Degrees of freedom, calculated as in (15), are along the top of each plot; at all but the smallest λ values, equal λ have higher df^t for higher γ since there is less shrinkage of $\hat{\beta}_j \neq 0$. This relationship switches only when df^t nears n, indicating that the heuristic in (15) might underestimate df for clearly over-fit models. Figure 4 shows CV and AICc error estimates and we see that the two criteria roughly track each other. Notice that for larger γ the CV error increases more quickly away from its minimum; this is predicted by Theorem 3.1 and shows that the consequences of over fit are worse with faster diminishing-bias. Computation times also increase with larger γ, although a single run at $\gamma = 10$ still requires less than a second.

Our simulation experiment includes `gamlr` runs of GL with γ of 0 (lasso), 1, and 10. For penalty selection, we focus on AICc with the GL *df* from Section 4 as well as 5-fold CV (with λ selected to *minimize* CV error estimates); results for additional selection rules are in the supplement. We also consider 'GL-select', a routine which chooses $\gamma \in \{0, 1, 10\}$ to minimize these AICc or CV values. Predictive performance is measured through the average root mean square error (RMSE) for $\hat{\mathbf{y}}$ on $\tilde{\mathbf{y}}$, and RMSE values are reported in terms of percentage-worse than the oracle-support MLE procedure. We include the oracle regression R^2 for reference.

We include performance results for MLE fit on 'oracle' restricted support. For the strictly sparse model, with $J = n/10$, our oracle uses the true nonzero support. For the strictly dense model, our oracle comparator is the C_p optimal ℓ_0 penalized solution

$$\boldsymbol{\beta}^\star = \underset{\boldsymbol{\beta}}{\operatorname{argmin}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + 2\sigma^2 \sum_j \mathbb{1}_{\{\beta_j \neq 0\}} \right\}, \tag{18}$$

which we solve by searching through OLS on $\mathbf{X}_{\{1\dots j\}}$ for $j = 1 \dots p$ (since the true coefficients are ordered). See Mallows (1973) and remarks after Theorem 3.1 for background on this oracle.

As a 'cheap' comparator, with run-time similar to that of GL, we consider the *marginal* adaptive lasso with ℓ_1 penalty weights $\omega_j \propto |\mathrm{cor}(\mathbf{x}_j, \mathbf{y})|^{-1}$. The AL weights are scaled so that $\min(\omega_j) = 1$ and we set *df* as the number of nonzero estimated parameters. As an 'expensive' gold standard, we include an exact solver for MCP penalized regression. Described in Section 3, `sparsenet` applies 5-fold CV to optimize out-of-sample error over a dense grid of potential penalty sizes (λ) and concavities (analogous to our γ) for the MCP penalty.

Table 1 presents a summary of predictive performance over all simulation models, Figure 5 plots some of the quantities from Theorem 3.1 for different algorithms, and Table 2 summarizes estimation error against true coefficients in our sparse simulation model. These results represent only a small portion of the simulation study. We have aggregated across different covariate designs (binary or continuous, with various levels of multicollinearity) and have combined results from $\kappa \in \{10, 50\}$ as 'fast' decay and those for $\kappa \in \{100, 200\}$ as 'slow' decay. The supplement contains an additional 128 tables, detailing hundreds of data generating processes and algorithm configurations, with results on prediction and estimation error, on the fitted number of nonzero parameters, and on sensitivity and false discovery rates.

	$\frac{\text{sd}(\boldsymbol{\eta})}{\sigma}$	**% Worse than Oracle RMSE** lasso *AICc*	lasso *CV*	GL $\gamma=1$ *AICc*	GL $\gamma=1$ *CV*	GL $\gamma=10$ *AICc*	GL $\gamma=10$ *CV*	GL-select *AICc*	GL-select *CV*	adapt. lasso *AICc*	adapt. lasso *CV*	MCP	*Oracle* R^2
dense model, fast decay													
	2	11	9	9	7	7	7	7	7	13	12	**6**	*0.78*
n=1000	*1*	9	8	8	7	9	8	8	7	8	8	**6**	*0.46*
	0.5	**4**	**4**	5	5	8	6	5	**4**	6	7	**4**	*0.15*
	2	51	46	38	54	13	61	19	47	27	**10**	46	*0.68*
n=100	*1*	12	12	12	14	16	15	14	13	**6**	12	12	*0.29*
	0.5	**0**	**0**	14	**0**	26	1	21	1	3	19	1	*0.00*
dense model, slow decay													
	2	23	**9**	17	10	10	21	10	**9**	20	16	10	*0.73*
n=1000	*1*	12	10	11	13	15	21	11	10	**9**	**9**	10	*0.37*
	0.5	**2**	3	4	3	4	3	3	3	3	4	3	*0.07*
	2	47	45	12	54	**0**	56	7	46	23	1	45	*0.59*
n=100	*1*	2	3	3	5	5	5	5	4	**-3**	1	3	*0.09*
	0.5	**-2**	**-2**	19	**-2**	23	**-2**	19	-1	1	17	**-2**	*-0.06*
sparse model, fast decay													
	2	10	9	8	7	6	6	7	7	12	12	**5**	*0.78*
n=1000	*1*	8	7	7	6	9	7	8	6	7	7	**5**	*0.45*
	0.5	**3**	**3**	**3**	**3**	6	4	**3**	**3**	4	5	**3**	*0.12*
	2	49	41	46	40	36	56	38	38	33	**27**	37	*0.77*
n=100	*1*	24	24	27	26	32	30	30	24	**18**	26	24	*0.44*
	0.5	**6**	**6**	14	7	33	7	27	7	9	26	7	*0.10*
sparse model, slow decay													
	2	14	12	10	9	6	5	6	5	17	17	**4**	*0.78*
n=1000	*1*	14	13	13	13	16	20	14	13	13	13	**12**	*0.45*
	0.5	**5**	**5**	6	6	7	7	**5**	**5**	6	7	**5**	*0.12*
	2	52	43	45	43	35	67	39	41	34	**28**	40	*0.77*
n=100	*1*	25	25	28	29	33	32	31	25	**19**	27	26	*0.44*
	0.5	**6**	7	18	7	33	7	28	7	9	26	7	*0.10*

Table 1: Out-of-sample predictive RMSE, reported as % worse than oracle (corresponding R^2 on far right), averaged over 1000 samples from various configurations of (17). The dense models have $J = p$ and the sparse models have $J = n/10$. Fast decay includes $\kappa \in \{10, 50\}$, while slow decay is $\kappa \in \{100, 200\}$. The oracle is MLE fit either on C_p-optimal support for the dense model or on the true support for the sparse model. Each row of this table corresponds to average performance across many data generating processes; see the supplement for more detailed results. Lasso (GL $\gamma = 0$), GL, and AL routines were executed in `gamlr`. MCP denotes results from the `sparsenet` MCP solver. GL-select chooses amongst $\gamma \in \{0, 1, 10\}$ using either AICc or CV. The best results are bolded.

	$\frac{\text{sd}(\boldsymbol{\eta})}{\sigma}$	**Estimation RMSE on true coefficients**									
		lasso		GL $\gamma = 1$		GL $\gamma = 10$		adapt. lasso			
		AICc	*CV*	*AICc*	*CV*	*AICc*	*CV*	*AICc*	*CV*	MCP	*Oracle*
sparse model											
	2	0.05	0.05	0.04	0.03	0.01	**0**	1.78	1.77	**0**	*0.00*
n=1000	*1*	0.12	0.12	**0.11**	**0.11**	0.13	0.12	3.41	3.41	**0.11**	*0.02*
	0.5	**0.15**	**0.15**	**0.15**	**0.15**	0.17	**0.15**	6.47	6.54	**0.15**	*0.10*
	2	0.05	0.04	**0.03**	0.04	0.04	0.06	1.07	1.02	0.04	*0.00*
n=100	*1*	**0.07**	0.08	0.08	0.08	0.12	0.08	1.89	2.02	0.08	*0.00*
	0.5	**0.08**	**0.08**	0.12	**0.08**	0.2	**0.08**	3.49	4.03	**0.08**	*0.02*

Table 2: Summary of estimation RMSE against the true coefficients from our *sparse* simulation model (where estimation of the true coefficients is a well-posed task). Results are averages over 1000 samples from each of the possible data generating process configurations, including both binary and continuous designs, ρ in $\{0, 0.5, .9\}$ and all of our decay values. The oracle is MLE fit on the true sparse support and the best results are bolded.

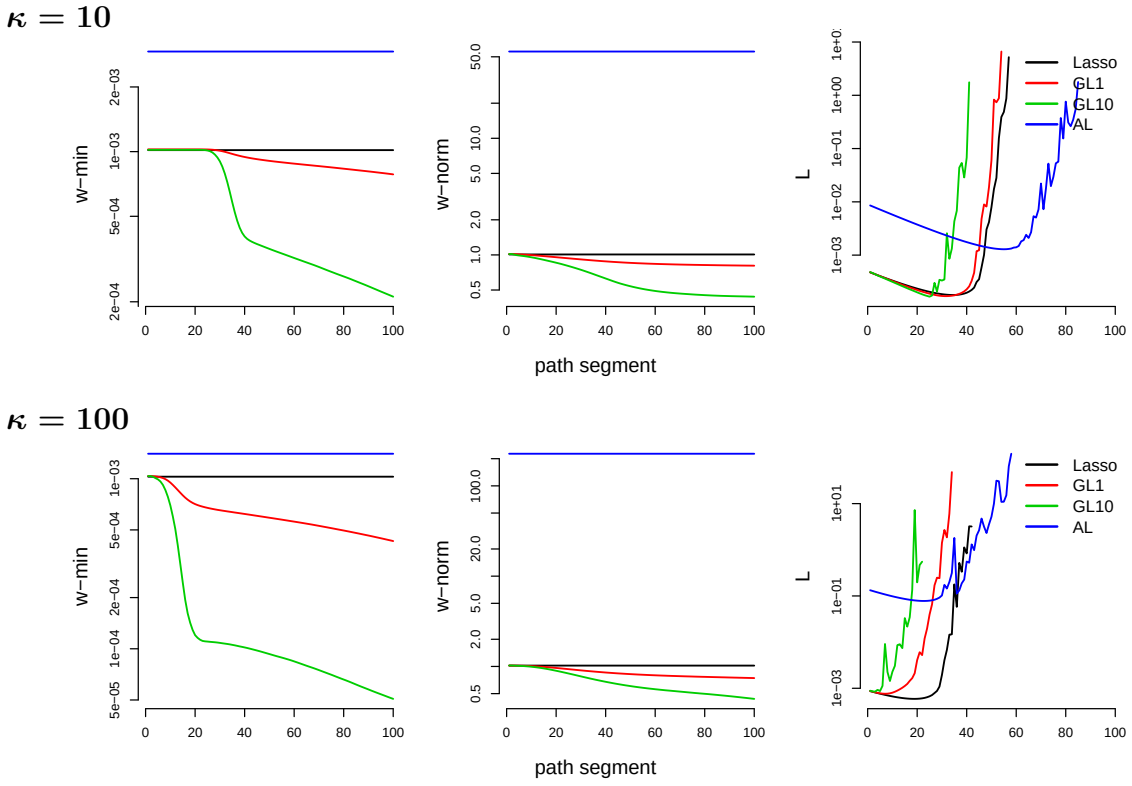

Figure 5: Properties of the weighted ℓ_1 penalized paths in our simulation, averaged across 1000 samples of $n = 1000$ for the *dense* model with *binary* design, $\rho = 0.9$, and $\text{sd}(\boldsymbol{\eta})/\sigma = 1$. The w-min values are $\omega^{\min}_{S^c}$, w-norm are $\|\boldsymbol{\omega}_S\|$, and L is the restricted eigenvalue constant; all as defined in Theorem 3.1. Note that, by construction, the λ grids are the same across all algorithms for the same data sample.

Remarks

Predictive performance

- AICc selection for one of the three GL γ-specifications is always able to provide predictive performance near to that of the computationally intensive routine of MCP (with CV selection over a grid in penalty size and scale). Looking at the lasso, $\gamma = 1$, and $\gamma = 10$ columns in Table 1, the best OOS RMSE is never more than 2% worse than MCP. Note that MCP is only a dominant performer when the sample size is large ($n = 1000$) and the true model is effectively sparse (either strictly sparse, or dense with fast decay).

- For $n = 1000$, a single `gamlr` run requires a fraction of a second; `sparsenet` with 5-fold CV usually required 15-20 seconds per run, and occasionally much longer. The AICc version of GL-select is within 2% of the MCP RMSE *except* when $n = 100$ and signal-to-noise (*s2n*) is 0.5 for the dense model or 0.5-1 for the sparse model. Thus AICc selection on three GL paths (potentially run in parallel), with combined compute time still a fraction of a second, yields a best or near-best predictor in all scenarios but for these small-data low-signal settings.

- CV GL-select and MCP provide very similar performance: their RMSE is always within 1% of each other, with MCP usually better at $n = 1000$ and GL-select usually better at $n = 100$. The combined cost of each GL CV routine is still far less than a single MCP run, and GL-select has the advantage that each path – across different folds and γ – can be run in parallel.

- The relationship between CV and AICc results for GL is dependent upon γ, signal strength, and sample size. For small $n = 100$ datasets with strong signal ($s2n = 2$), AICc does generally better than CV; e.g., this is the one setting where AICc GL-select outperforms MCP by a large margin. For $n = 100$ samples with weak signal ($s2n \leq 1$), the opposite is true and CV outperforms AICc if $\gamma > 0$. With larger $n = 1000$ samples, AICc and CV perform more similarly *except* in the effectively dense model.

In summary, CV and AICc give similar results whenever n/s is large (regardless of p). CV is safer when n/s is small and there is little signal (in this situation even the oracle yields small or negative R^2), while AICc is best if n/s is small but there is strong signal. The supplement shows that AICc is almost always a massive improvement over either BIC or AIC for prediction.

Estimation performance and information compression

- Table 2 shows mostly similar estimation errors between the GL and MCP comparators. MCP dominates only in the $n = 1000$ samples with strong $s2n = 2$ signal. The simple lasso is also competitive, despite its non-diminishing bias, in all but the larger samples with $s2n \geq 1$.

- Marginal AL performed well in prediction, with RMSE that was near to that of the lasso when $n = 1000$ and sometimes much better than MCP or GL when $n = 100$. However, Table 2 shows *terrible* performance in terms of estimation error for marginal AL. It is interesting that marginal AL outperforms MCP in prediction when $n = 100$ and $s2n \geq 1$, but has estimation RMSE that is an order of magnitude larger in the same scenario.

- In the supplement, CV and AICc selected GL1 has usually 20-80% the number of nonzero coefficients as the corresponding lasso fit. GL10 leads to even more sparsity, often returning less than 10% of the selected *df* from lasso. This suggest `gamlr` with a small but nonzero γ (e.g., as in GL1) as a reliable strategy for compressing information without hurting predictive performance (our original motivating goal). The supplement also shows that $\gamma > 0$ leads to a large drop in false discovery (with respect to oracle support) relative the the lasso. In contrast, marginal AL yields no more (and often less) sparsity than the standard lasso and provides none of the desired information compression gain. Overall, MCP seems to have best variable selection properties (reducing FDR without dramatically lower sensitivity).

Path properties

- Figure 5 shows how the quantities in Theorem 3.1 behave for various coefficient decay rates in a highly collinear ($\rho = .9$) and moderately noisy ($s2n = 1$) setting. Recall that prediction error should get closer to that of an ℓ_0 oracle if $\|\boldsymbol{\omega}_S\|$ – the norm on the weights in the support of the ℓ_0 rule – can be made small without $\omega^{\min}_{S^c}$ – the smallest weight on the complement of this support – shrinking too much. In comparison to lasso, this is achieved for $\gamma = 1$ under fast coefficient decay ($\kappa = 10$, high effective sparsity) while $\omega^{\min}_{S^c}$ drops right after the path begins under slower decay and lower effective sparsity. For the fast diminishing bias of $\gamma = 10$, $\omega^{\min}_{S^c}$ drops more dramatically and earlier; only in the $\kappa = 10$ setting does there appear to be any opportunity for improved prediction from $\gamma = 10$ relative to $\gamma = 1$. Marginal AL provides an interesting side-case; it maintains higher $\omega^{\min}_{S^c}$ at the expense of a *much* higher $\|\boldsymbol{\omega}_S\|$.

• The far right panels in Figure 5 show the realized value of $L = \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}} \left(\omega_{S^c}^{\min} - \sqrt{2}\sigma/\lambda\right)^{-1}$, the constant governing our restricted eigenvalue $\phi^2(L, S)$ at the C_p-optimal model (see Definition 3.1). A small value of this constant is important for keeping prediction error low (by keeping $\phi^2(L, S)$ big). Moreover, in the supplement we show that small values of L are essential in controlling the false discovery rate. In Figure 5, we see that L is decreasing steadily for each algorithm until the point where $\omega_{S^c}^{\min}$ drops; after this point it explodes and, soon after that, becomes undefined when $\omega_{S^c}^{\min} < \sqrt{2}\sigma/\lambda$.

6 Hockey example

We close with an example analysis: measuring the performance of hockey players. It extends analysis in Gramacy et al. (2013, 2015). The data include every goal in the National Hockey League (NHL) back to the 2002-2003 season: 69,449 goals and 2439 players.

For goal i in season s with away team a and home team h, say that q_i is the probability that the home team scored this goal. Our regression model is then

$$\text{logit}\,[q_i] = \alpha_0 + \alpha_{sh} - \alpha_{sa} + \mathbf{u}_i'\boldsymbol{\phi} + \mathbf{x}_i'\boldsymbol{\beta}_0 + \mathbf{x}_i'\boldsymbol{\beta}_s, \tag{19}$$

Vector $\mathbf{u}_i$ holds indicators for various special-teams scenarios (e.g., a home team power play), and $\boldsymbol{\alpha}$ provides matchup/season specific intercepts. Vector $\mathbf{x}_i$ contains player effects: $x_{ij} = 1$ if player j was on the home team and on ice for goal i, $x_{ij} = -1$ for away player j on ice for goal i, and $x_{ij} = 0$ for everyone not on the ice. Coefficient $\beta_{0j} + \beta_{sj}$ is the season-s effect of player j on the log odds that, given a goal has been scored, the goal was scored by their team. These effects are 'partial' in that they control for who else was on the ice, special teams scenarios, and team-season fixed effects – a player's β_{0j} or β_{sj} only need be nonzero if that player effects play above or below the team average for a given season.

We estimate GL paths of $\hat{\boldsymbol{\beta}}$ from (19) with $\boldsymbol{\alpha}$ and $\boldsymbol{\phi}$ left *unpenalized*. Coefficient costs are *not* scaled by covariate standard deviation, since this would have favored players with little ice time. Joint $[\gamma, \lambda]$ surfaces for AICc and BIC are in Figure 6. AICc favors denser models with low λ but not-to-big γ, while the BIC prefers very sparse but relatively unbiased models with large λ and small γ. Both criteria are strongly adverse to any model above $\gamma = 100$, which

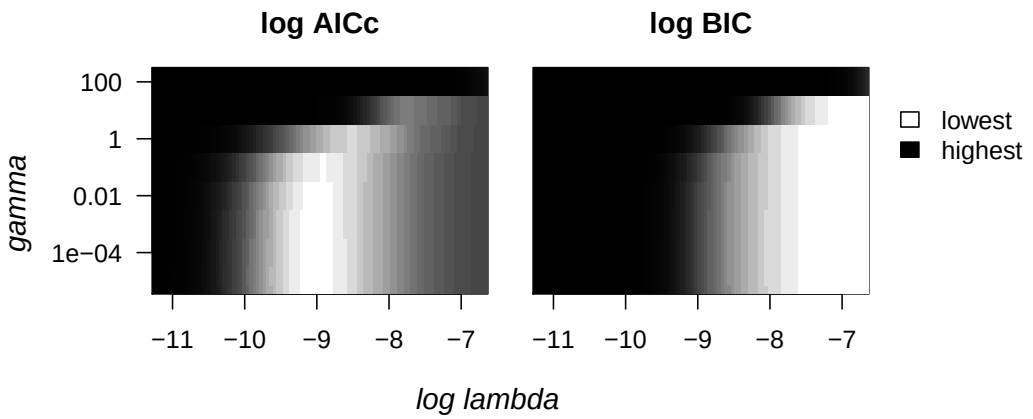

Figure 6: Hockey example AICc and BIC surfaces, rising from white to black on log scale.

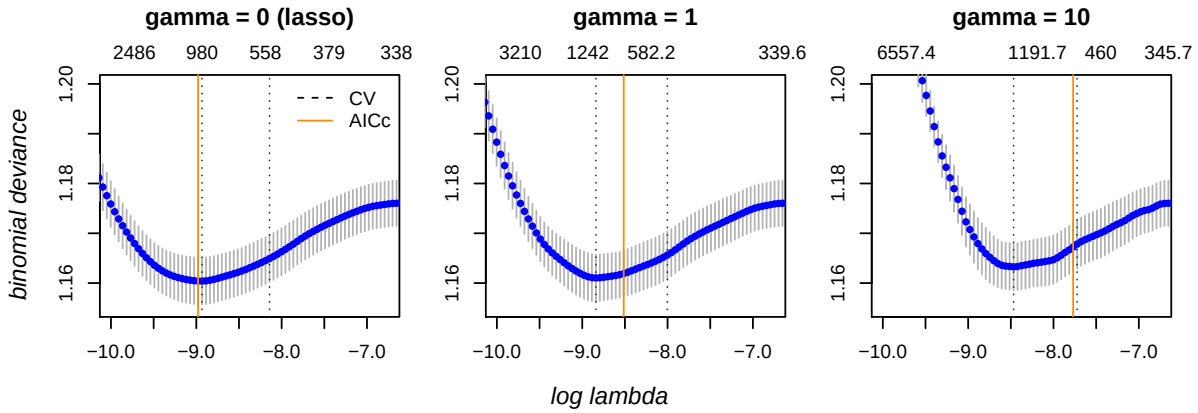

Figure 7: Hockey example 10-fold CV: mean OOS deviance $\pm$1se, with minimum-error and 1SE selection rules marked with black dotted lines, and solid orange line showing AICc selection.

is also where timings explode (supplement). Ten-fold CV results are shown in Figure 7 for γ of 0, 1, and 10. The OOS error minima are around the same in each case – average deviance slightly above 1.16 – but errors increase much faster away from optimality with larger γ. AICc selection is always between the CV error-minimizing selection and that of the common *1SE* rule (largest λ with mean OOS error no more than 1 standard error away from the minimum).

The original goal with this dataset was to build a better version of hockey's 'plus-minus' (PM) statistic: number of goals *for* minus *against* each player's team while he is on the ice. To convert from player effects $\beta_{0j} + \beta_{sj}$ to the scale of 'plus/minus', set the probability that a goal was scored by his team given player j is on ice (and no other information) as $p_j = e^{\beta_j}/(1+e^{\beta_j})$.

	lasso			$\gamma = 1$			$\gamma = 10$		
		PPM	PM		PPM	PM		PPM	PM
1	Ondrej Palat	33.8	38	Sidney Crosby	29.2	52	Sidney Crosby	32.6	52
2	Sidney Crosby	31.2	52	Ondrej Palat	29	38	Jonathan Toews	22.8	35
3	Henrik Lundqvist	25.8	9	Jonathan Toews	21.4	35	Joe Thornton	22	34
4	Jonathan Toews	24	35	Joe Thornton	21	34	Anze Kopitar	22	39
5	Andrei Markov	23.1	34	Andrei Markov	20.9	34	Andrei Markov	20.7	34
6	Joe Thornton	21.4	34	Henrik Lundqvist	19.8	9	Alex Ovechkin	18.1	16
7	Anze Kopitar	20.6	39	Anze Kopitar	19.5	39	Pavel Datsyuk	16.6	13
8	Tyler Toffoli	18.9	31	Pavel Datsyuk	16.1	13	Ryan Getzlaf	15.8	16
9	Pavel Datsyuk	17.7	13	Logan Couture	15.9	29	Henrik Sedin	15.2	7
10	Ryan Nugent-hopkins	17.4	18	Alex Ovechkin	15.8	16	Marian Hossa	14.9	21
11	Gabriel Landeskog	16.6	36	Marian Hossa	14.4	21	Alexander Semin	14.7	-1
12	Logan Couture	16.5	29	Alexander Semin	14.2	-1	Jaromir Jagr	14.5	28
13	Alex Ovechkin	15.8	16	Matt Moulson	13.9	22	Logan Couture	14.2	29
14	Marian Hossa	15.4	21	Tyler Toffoli	13.3	31	Matt Moulson	13.7	22
15	Alexander Semin	14.8	-1	David Perron	12.7	2	Mikko Koivu	13	12
16	Zach Parise	14.7	21	Mikko Koivu	12.5	12	Joe Pavelski	12.6	33
17	Frans Nielsen	13.5	8	Frans Nielsen	12.3	8	Steven Stamkos	12.6	24
18	Mikko Koivu	13.4	12	Ryan Getzlaf	12.1	16	Frans Nielsen	12.5	8
19	Matt Moulson	13.4	22	Ryan Nugent-hopkins	11.9	18	Marian Gaborik	12.3	29
20	David Perron	13.1	2	Jaromir Jagr	11.8	28	Zach Parise	12.2	21
		305 nonzero effects			*204 nonzero effects*			*64 nonzero effects*	

Table 3: Top 20 AICc selected player 'partial plus-minus' (PPM) values for the 2013-2014 season, under $\gamma = 0, 1, 10$. The number of nonzero player effects for each γ are noted along the bottom.

The 'partial plus/minus' (PPM) is

$$\text{ppm}_j = N_j(p_j - (1 - p_j)) = N_j(2p_j - 1) \tag{20}$$

where N_j is the number of goals for which he was on-ice. This measures quality and quantity of contribution and lives on the same scale as PM. See Gramacy et al. (2015) for details.

Table 3 contains the estimated PPM values for the 2013-2014 season under various γ levels, using AICc selection. We see that, even if changing concavity (γ) has little effect on minimum CV errors (Figure 7), larger γ yield more sparse models and different conclusions about player contribution. At the $\gamma = 0$ lasso, there are 305 nonzero player effects (individuals measurably different from their team's average ability) and the list includes young players who have had very strong starts to their careers. For example, Ondrej Palat and Tyler Toffoli both played their first full seasons in the NHL in 2013-2014. As γ increases to 1, these young guys drop in rank while more proven stars (e.g., Sidney Crosby and Jonathan Toews) move up the list. Finally, at $\gamma = 10$ only big-name stars remain amongst the 64 nonzero player effects.

7 Discussion

Whenever exact solvers are too expensive, concave penalized estimation reduces largely to weighted-ℓ_1 penalization. Path adaptation is an intuitively reasonable source of weights, and we are able to show that POSE – particularly `gamlr` with AICc selection – provides high quality diminishing-bias sparse regression at *no more cost* than a single lasso path. We know of no other software that meets this standard.

References

Akaike, H. (1973). Information theory and the maximum likelihood principle. In B. Petrov and F. Csaki (Eds.), *2nd International Symposium on Information Theory*, Akademiai Kiado, Budapest.

Armagan, A., D. B. Dunson, and J. Lee (2013). Generalized Double Pareto Shrinkage. *Statistica Sinica 23*, 119–143.

Bickel, P. J. (1975). One-Step Huber Estimates in the Linear Model. *Journal of the American Statistical Association 70*, 428–434.

Bickel, P. J., Y. Ritov, and A. B. Tsybakov (2009). Simultaneous analysis of Lasso and Dantzig selector. *The Annals of Statistics 37*(4), 1705–1732.

Breheny, P. and J. Huang (2011, March). Coordinate descent algorithms for nonconvex penalized regression, with applications to biological feature selection. *The Annals of Applied Statistics 5*(1), 232–253.

Breiman, L. (1996). Heuristics of instability and stabilization in model selection. *The Annals of Statistics 24*(6), 2350–2383.

Bühlmann, P. and S. van de Geer (2011). *Statistics for High-Dimensional Data*. Springer.

Candes, E. J., M. B. Wakin, and S. P. Boyd (2008). Enhancing Sparsity by Reweighted L1 Minimization. *Journal of Fourier Analysis and Applications 14*, 877–905.

Cevher, V. (2009). Learning with Compressible Priors. In *Neural Information Processing Systems (NIPS)*.

Efron, B. (2004). The estimation of prediction error: covariance penalties and cross-validation. *Journal of the American Statistical Association 99*, 619–632.

Efron, B., T. Hastie, I. Johnstone, and R. Tibshirani (2004). Least Angle Regression. *Annals of Statistics 32*, 407–499.

Fan, J. and R. Li (2001). Variable Selection via Nonconcave Penalized Likelihood and its Oracle Properties. *Journal of the American Statistical Association 96*, 1348–1360.

Fan, J. and H. Peng (2004). Nonconcave Penalized Likelihood with a Diverging Number of Parameters. *The Annals of Statistics 32*, 928–961.

Fan, J., L. Xue, and H. Zou (2014). Strong oracle optimality of folded concave penalized estimation. *The Annals of Statistics 42*(3), 819–849.

Flynn, C., C. Hurvich, and J. Simonoff (2013). Efficiency for regularization parameter selection in penalized likelihood estimation of misspecified models. *Journal of the American Statistical Association 108*, 1031–1043.

Friedman, J., T. Hastie, and R. Tibshirani (2010). Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software 33*, 1–22.

Friedman, J. H. (2008). Fast Sparse Regression and Classification. Technical Report, Dept. of Statistics, Stanford University.

Gentzkow, M., J. Shapiro, and M. Taddy (2015). Measuring polarization in high dimensional data. *Chicago Booth working paper*.

Gramacy, R., S. Jensen, and M. Taddy (2013). Estimating player contribution in hockey with regularized logistic regression. *Journal of Quantitative Analysis in Sports 9*, 97–111.

Gramacy, R., M. Taddy, and S. Tian (2015). Hockey performance via regression. *to appear in the Handbook of statistical methods for design and analysis in sports*.

Hoerl, A. and R. Kennard (1970). Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics 12*, 55–67.

Huang, J., S. Ma, and C.-H. Zhang (2008). Adaptive Lasso for sparse high-dimensional regression models. *Statistica Sinica 18*(4), 1603.

Hurvich, C. M. and C.-L. Tsai (1989). Regression and time series model selection in small samples. *Biometrika 76*(2), 297–307.

Mallows, C. L. (1973). Some comments on CP. *Technometrics 15*, 661–675.

Mazumder, R., J. H. Friedman, and T. Hastie (2011). SparseNet : Coordinate Descent With Nonconvex Penalties. *Journal of the American Statistical Association 106*, 1125–1138.

Park, T. and G. Casella (2008). The Bayesian Lasso. *Journal of the American Statistical Association 103*, 681–686.

Raskutti, G., M. J. Wainwright, and B. Yu (2010). Restricted eigenvalue properties for correlated Gaussian designs. *The Journal of Machine Learning Research 11*, 2241–2259.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*, 461–464.

Stein, C. M. (1981). Estimation of the mean of a multivariate normal distribution. *The Annals of Statistics 9*, 1135–1151.

Taddy, M. (2013). Multinomial inverse regression for text analysis. *Journal of the American Statistical Association 108*, 755–770.

Taddy, M. (2015). Distributed multinomial regression. *The Annals of Applied Statistics*. To appear.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society, Series B 58*, 267–288.

Wainwright, M. J. (2006). Sharp thresholds for high-dimensional and noisy recovery of sparsity. *UC Berkeley Technical Report*.

Wainwright, M. J. (2009). Sharp thresholds for high-dimensional and noisy sparsity recovery using L1-constrained quadratic programming (Lasso). *IEEE Transactions on Information Theory 55*, 2183–2202.

Wang, L., Y. Kim, and R. Li (2013, October). Calibrating nonconvex penalized regression in ultra-high dimension. *The Annals of Statistics 41*(5), 2505–2536.

Zhang, C.-H. (2010a). Nearly unbiased variable selection under minimax concave penalty. *Annals of Statistics 38*, 894–942.

Zhang, T. (2010b). Analysis of multi-stage convex relaxation for sparse regularization. *The Journal of Machine Learning Research 11*, 1081–1107.

Zhang, T. (2013). Multi-stage convex relaxation for feature selection. *Bernoulli 19*, 2277–2293.

Zou, H. (2006). The adaptive lasso and its oracle properties. *Journal of the American Statistical Association 101*, 1418–1429.

Zou, H. and T. Hastie (2005). Regularization and variable selection via the elastic net. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 67*(2), 301–320.

Zou, H., T. Hastie, and R. Tibshirani (2007). On the degrees of freedom of the lasso. *The Annals of Statistics 35*, 2173–2192.

Zou, H. and R. Li (2008). One-step sparse estimates in nonconcave penalized likelihood models. *The Annals of Statistics 36*, 1509–1533.