One-step estimator paths for concave regularization

Matt Taddy

The University of Chicago Booth School of Business
faculty.chicagobooth.edu/matt.taddy

This article explores some practical and computational issues related to the application of sparse diminishing-bias regularized regression: techniques which mimic estimation under coefficient penalty functions spanning the range of concavity between L_0 and L_1 norms. In Big data applications, where exact solvers are too computationally expensive, such techniques reduce largely to L_1 regularization with data-dependent weights. We present novel results on the distance between weighted-L_1 penalized minimization and L_0 minimization, giving some intuition on the amount of information compression that is possible. We then propose a simple new algorithm framework – the path of one-step estimators (POSE) – which 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. POSE provides the benefits of diminishing-bias, but at a computational cost close or equal to that of the standard lasso. One specific implementation of POSE, the 'gamma lasso', is motivated from Bayesian foundations and we work through the details necessary for its application. The methods and issues discussed are illustrated through use of the gamma lasso in extensive linear regression 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 that of 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 the 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 estimation has developed over the past 15 years. For example, many authors (e.g., from Fan and Li 2001 and 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'.

Unfortunately, exact solvers for concave penalized estimation all require significantly more compute time than a standard lasso. As we review in Section 2.1, recent literature recommends the use of approximate solvers. Certainly, this is necessary for data of the size we encounter in analysis of, say, internet commerce. These approximations take the form of iteratively-weighted-L_1 regularization, where the coefficient-specific weights are based upon pre-estimates of the coefficients taken from previous iterations of the approximate solver. One theme of this literature (e.g., Zou and Li, 2008; Fan et al., 2014) holds that even a single step of 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 with such one-step estimation (OSE) is finding starts that are, indeed, good enough.

This article presents a computational strategy for obtaining a *path* of one-step estimates under concave penalization. The generic POSE – path of one-step estimators – algorithm simply uses each solution along a regularization path as the basis for weights at the next path iteration. In this way, we are able to achieve near-lasso speeds while obtaining a path of sparse diminishing-bias estimators. Our implementation and illustrations apply POSE using the log penalty (e.g. Candes et al., 2008), and we refer to this as the 'gamma lasso' for its interpretation as the posterior mode under a hierarchical Bayesian model. However, we make no claims for

theoretical superiority for one target penalty over another. Rather, we treat one-step weighted-L_1 estimators at face-value and address practical issues in computation for such schemes:

- minimizing the cost of computing good pre-estimates,
- moving quickly along a path of candidate estimators, and
- having tools available for selection amongst these candidates.

All three issues are related to each other, and we will illustrate their interdependency.

Section 2 is a survey of sparse diminishing-bias regularization: we review the connection between concave and weighted-L_1 penalties in 2.1, and present novel results on the distance between weighted-L_1 and L_0 minimization in 2.2. The POSE algorithm is introduced in Section 3, with our gamma lasso specification in 3.1 and its motivation from Bayesian foundations in 3.2. Section 4 covers implementation issues key to any path-based strategy: speed and stability in 4.1, and model selection in 4.2. Finally, Section 5 presents two empirical studies: a simulation experiment in 5.1, and in 5.2 we investigate the data analysis question: given all goals in the past decade of NHL hockey, what can we say about individual player contributions?

Throughout, we focus on practice and computation. For example, the theory of Section 2.3 and the experiment in Section 5.1 take L_0 minimization as the oracle comparator. We are agnostic about existence of any sparse 'truth' and promote these techniques from an information-compression perspective. And Section 4 emphasizes quick selection amongst a range of penalty sizes, as required by any practitioner who is unwilling to rely upon theoretically optimal specification when analyzing real data. The goal is to help nudge sparse diminishing-bias regularization into the Big Data mainstream.[1]

2 Sparse regularization paths and diminishing bias

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 $\boldsymbol{x}_j = [x_{1j}, \ldots, x_{nj}]'$.[2] Write

[1] To this end, we also provide polished and well-documented software in the `gamlr` package for R.

[2] Since the size of penalized β_j depends upon the units of x_{ij}, it is common to scale the coefficient by $\mathrm{sd}(\boldsymbol{x}_j)$, the standard deviation of the j^{th} column of $\mathbf{X}$; this is achieved if x_{ij} is replaced by $x_{ij}/\mathrm{sd}(\boldsymbol{x}_j)$ throughout.

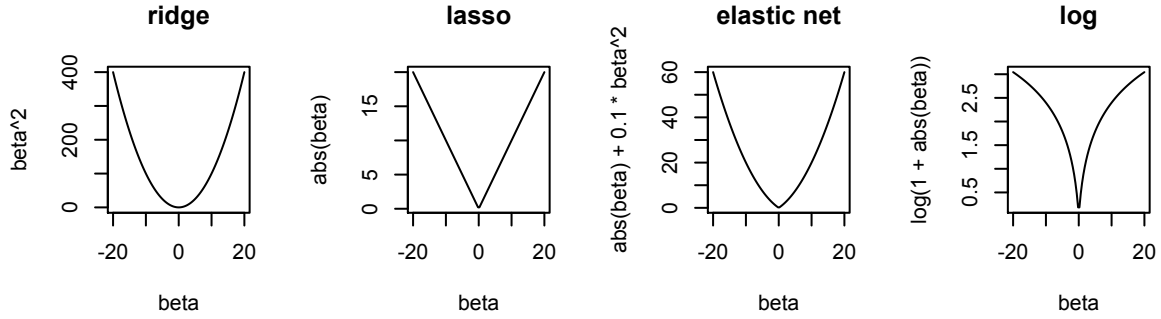

Figure 1: From left to right, L_2 costs (ridge, Hoerl and Kennard, 1970), L_1 (lasso, Tibshirani, 1996), the 'elastic net' mixture of L_1 and L_2 (Zou and Hastie, 2005), and the log penalty (Candes et al., 2008).

$\eta_i = \alpha + \mathbf{x}_i'\boldsymbol{\beta}$ as the linear equation for observation i, and denote with $l(\alpha, \boldsymbol{\beta}) = l(\boldsymbol{\eta})$ an unregularized objective proportional to the negative log likelihood. 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 then the solution to

$$\min \left\{ l(\alpha, \boldsymbol{\beta}) + n\lambda \sum_{j=1}^{p} c_j(|\beta_j|) \right\}, \tag{1}$$

where $\lambda > 0$ controls overall penalty magnitude and are $c_j()$ are coefficient cost functions.

A few common cost functions are shown in Figure 1. 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, which span the range between L_1 and L_0 penalties.

The 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 obtained while moving from high to low penalization along $\lambda^1 > \lambda^2 \ldots > \lambda^T$ (e.g., LARS in Efron et al., 2004, is a well known example). These paths begin at λ^1 set to infimum λ such that (1) is minimized at $\hat{\boldsymbol{\beta}} = \mathbf{0}$ (see Appendix A), and proceed down to some pre-specified λ^T (e.g., $\lambda^T = 0.01\lambda^1$).

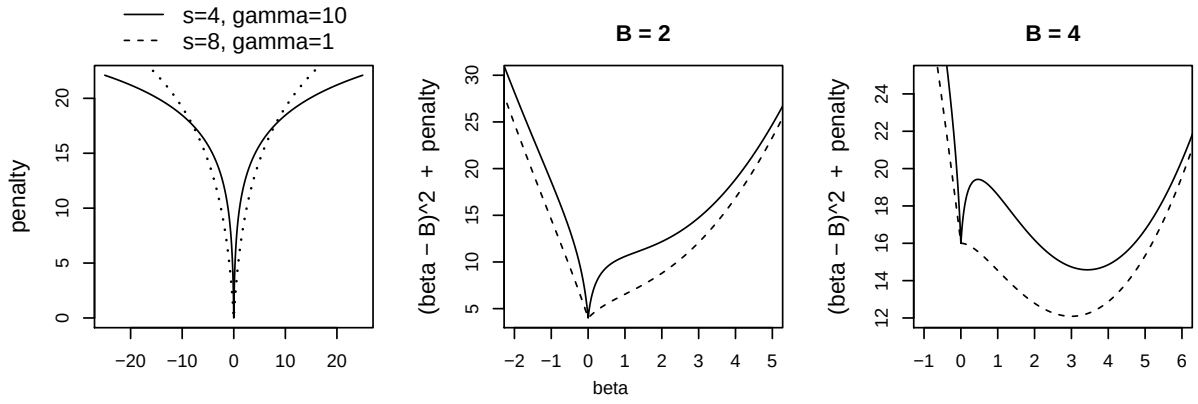

Figure 2: Log penalties $c(\beta) = s\log(1+\gamma|\beta|)$ and penalized objectives $(\beta - B)^2 + c(\beta)$.

2.1 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 in the introduction. 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 concave minimization objective; see Figure 2 for illustration.

This concavity makes computation difficult. For example, one run of SCAD via the `ncvreg` R package (Breheny and Huang, 2011) for the simulation in Section 5.1 requires around 10 minutes, compared to 1-2 seconds for the lasso. The slowness is due to the difficulty of concave minimization: finding global, or even good local, minima is an extremely tough task. 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 under decreasing L_1 penalty magnitude and, for each segment on this path, adapts coefficient estimates along a second path of increasing penalty concavity. While far more efficient than previous alternatives, `sparsenet` relies upon solution over a large set of penalty specifications[3] and its compute cost remains much higher than for lasso (e.g., 15-20 seconds in our simulation).

A relatively fast class of solvers for concave penalized estimators uses local linear approximation (LLA; e.g., Candes et al., 2008). LLA replaces the concave cost function c_j with its tangent at the current estimate, $c_j'(\hat{\beta}_j)\beta_j$. The objective is then just a weighted L_1 penalized

[3]POSE shares with `sparsenet` the idea of moving along a path of closely related specifications, but does not require a grid in both cost size and concavity. Intuitively, POSE runs a path diagonally through this grid.

loss (solvable, say, as in Appendix D), and one 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 does well in practice. Importantly, they show that LLA does well even (or especially) if you *stop it after one iteration*. This is an example of one-step estimation (OSE), a technique inspired by Bickel (1975) that amounts to taking as your estimator the first step of an iterative approximation to some objective. Such early-stopping can be as good as the full-step 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. He shows that just one or two steps here is sufficient for obtaining oracle support recovery properties under much weaker conditions than required by a standard lasso. 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 regression coefficients $\hat{\beta}_j^0 = \mathrm{cor}(\mathbf{x}_j, \mathbf{y})$.[4]

OSE LLA, or a two-step estimator starting from $\hat{\boldsymbol{\beta}} = 0$ as suggested in Fan et al. (2014) and Wang et al. (2013), or any version of the adaptive lasso, are *all just weighted-L_1 minimization*. These methods differ only in how the weights are constructed. Regardless of which you prefer, weighted-L_1 penalties are likely to play a central role in diminishing-bias penalization whenever the dataset is too large for use of exact concave penalty solvers.

2.2 Comparison between weighted-L_1 and L_0 penalized estimation

Sparse estimation will be useful for (and often used in) settings where the true data generating process is non-sparse. This can be motivated from an information compression perspective

[4]We include marginal AL in our study of Section 5.1.

– e.g., in Big data settings such as Taddy (2013a), where many estimates need to be quickly communicated across multiple machines – or from a simple desire to minimize complexity and focus decision making. We are thus interested in L_1 penalization with data-dependent weights as a way to obtain fits that are as sparse as possible without compromising predictive ability.

Our oracle benchmark is estimation under L_0 costs, $c_j(\beta_j) = \mathbb{1}_{\{\beta_j \neq 0\}}$, for which global solution is practically impossible. We present finite sample bounds on the distance between L_0 and weighted L_1 penalized estimation. This comparison yields simple relationships that make no assumptions about the underlying data model. Moreover, when we do make standard assumptions about the data generating process, we are able to quantify distance between weighted-L_1 estimates and a theoretically optimal L_0 penalized estimator.

2.2.1 Sparse Approximation for Prediction

For any support subset $S \subset \{1 \dots p\}$ with cardinality $|S| = s$ and complement $S^c = \{1 \dots p\} \setminus S$, denote vectors restricted to 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 hat (projection) matrix from OLS restricted to S. We suppress intercepts throughout, and use $|\cdot|$ and $\|\cdot\|$ applied to vectors as the L_1 and L_2 norms, respectively.

We'll use the following simple result for stagewise regression – iterative fitting of new covariates to the residuals of an existing linear model (as in, e.g., Goldberger 1961).

LEMMA 2.1. *Say* $\mathrm{MSE}_S = \|\mathbf{X}\boldsymbol{\beta}^S - \mathbf{y}\|^2/n$ *and* $\mathrm{cov}(\boldsymbol{x}_j, \mathbf{e}^S) = \boldsymbol{x}'_j(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^S)/n$ *are sample variance and covariances. Then for any* $j \in 1 \dots p$*,*

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

Proof. From the well-known property on the correlation coefficient (R^2) for linear models, in-sample correlation and variances are such that

$$\frac{\mathrm{cov}^2(\boldsymbol{x}_j, \mathbf{e}^S)}{\mathrm{var}(\boldsymbol{x}_j)\mathrm{var}(\mathbf{e}^S)} = 1 - \frac{\mathrm{var}(\mathbf{e}^S - \tilde{\beta}_j\boldsymbol{x}_j)}{\mathrm{var}(\mathbf{e}^S)}$$

where $\tilde{\beta}_j = \boldsymbol{x}_j' \mathbf{e}^S / (\boldsymbol{x}_j' \boldsymbol{x}_j)$ is the stagewise coefficient estimate. Since $\mathrm{var}(\boldsymbol{x}_j) = 1$, multiplying everything by $\mathrm{var}(\mathbf{e}^S)$ yields $\mathrm{cov}^2(\boldsymbol{x}_j, \mathbf{e}^S) = \mathrm{var}(\mathbf{e}^S) - \mathrm{var}(\mathbf{e}^S - \tilde{\beta}_j \boldsymbol{x}_j) \leq \mathrm{var}(\mathbf{e}^S) - \mathrm{var}(\mathbf{e}^{S \cup j})$. The last inequality holds because $\mathbf{e}^{S \cup j}$, residuals from OLS on $\mathbf{X}_{S \cup j}$, have the smallest-possible sum of squares for that set of covariates. With $\mathrm{var}(\mathbf{e}^S) = \mathrm{MSE}_S$, etc, we are done. □

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

DEFINITION 2.1. *The restricted eigenvalue is* $\phi^2(L, S) = \min_{\{\mathbf{v}: \mathbf{v} \neq \mathbf{0},\ |\mathbf{v}_{S^c}| \leq L\sqrt{s}\|\mathbf{v}_S\|\}} \frac{\|\mathbf{X}\mathbf{v}\|^2}{n\|\mathbf{v}\|^2}$.

The specific RE in Definition (2.1) matches the 'adaptive restricted eigenvalues' of Buhlmann and van de Geer (2011). These and related RE quantities are common in prediction and estimation results for both L_1 and concave penalization schemes. See the remarks for more detail.

Finally, our result derives a bound on the distance between prediction rules based on L_0 and weighted-L_1 penalized estimation. Intercepts are suppressed for simplicity.

THEOREM 2.1. *Consider squared-error loss* $l(\boldsymbol{\beta}) = \frac{1}{2}\|\mathbf{X}\boldsymbol{\beta} - \mathbf{y}\|^2$*, and suppose* $\boldsymbol{\beta}^\nu$ *minimizes the* L_0 *penalized objective* $l(\boldsymbol{\beta}) + n\nu \sum_{j=1}^p \mathbb{1}_{\{\beta_j \neq 0\}}$ *with* $\boldsymbol{\beta}^\nu = \boldsymbol{\beta}^S$ *and* $|S| = s < n$*. 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* $\phi^2(L, S) > 0$ *implies*

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

with the restricted eigenvalue defined for $L = \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}} \left(\omega_{S^c}^{\min} - \sqrt{2\nu}/\lambda\right)^{-1}$.

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

$$\begin{aligned} \frac{1}{2}\|\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{y}\|^2 + n\lambda \sum_j \omega_j |\hat{\beta}_j| &= \frac{1}{2}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu)\|^2 + \frac{1}{2}\|\mathbf{e}^S\|^2 - \hat{\mathbf{y}}'\mathbf{e}^S + n\lambda \sum_j \omega_j |\hat{\beta}_j| \qquad (3) \\ &\leq \frac{1}{2}\|\mathbf{X}\boldsymbol{\beta}^\nu - \mathbf{y}\|^2 + n\lambda \sum_j \omega_j |\beta_j^\nu| = \frac{1}{2}\|\mathbf{e}^S\|^2 + n\lambda \sum_{j \in S} \omega_j |\beta_j^S| \end{aligned}$$

Since $\hat{\mathbf{y}}'\mathbf{e}^S = \hat{\mathbf{y}}'(\mathbf{I} - \mathbf{H}^S)\mathbf{y} = \hat{\boldsymbol{\beta}}'\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\nu) = \sum_{j \in S^c} \hat{\beta}_j \boldsymbol{x}_j'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\nu)$, we can apply Lemma 2.1 followed by $\boldsymbol{\beta}^\nu$ being optimal under L_0 penalty ν to get

$$\left(\frac{\boldsymbol{x}_j'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\nu)}{n}\right)^2 \leq \mathrm{MSE}_S - \mathrm{MSE}_{S \cup j} < 2\nu \quad \forall\, j \tag{4}$$

so that $|\hat{\mathbf{y}}'\mathbf{e}^S| = |\hat{\boldsymbol{\beta}}_{S^c}\mathbf{X}'_{S^c}(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\nu)| < n\sqrt{2\nu}|\hat{\boldsymbol{\beta}}_{S^c}|$. Applying this inside (3),

$$\frac{1}{2}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu)\|^2 + n\left(\omega_{S^c}^{\min}\lambda - \sqrt{2\nu}\right)|\hat{\boldsymbol{\beta}}_{S^c}| \;\leq\; n\lambda\sum_{j\in S}\omega_j|\hat{\beta}_j - \beta_j^\nu| \;\leq\; n\lambda\|\boldsymbol{\omega}_S\|\|\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^\nu\|. \tag{5}$$

Given $\omega_{S^c}^{\min}\lambda > \sqrt{2\nu}$, difference $\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu$ is in the RE support for $L = \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}}(\omega_{S^c}^{\min} - \sqrt{2\nu}/\lambda)^{-1}$ and thus $\|\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^\nu\| \leq \|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu)/\sqrt{n}\|/\phi(L,S)$. Finally, applying this inside (5) yields

$$\frac{1}{2}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu)\|^2 \;\leq\; \frac{\sqrt{n}\lambda\|\boldsymbol{\omega}_S\|\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu)\|}{\phi(L,S)}. \tag{6}$$

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

Remarks

- This is finite sample exact and completely non-parametric – it makes no reference to the true distribution of $\mathbf{y} \mid \mathbf{X}$. Indeed, if we make such assumptions, Theorem 2.1 provides bounds on the distance between a weighted-lasso and optimal prediction. The next remark is an example.

- If we assume that $\mathbf{y} \sim (\boldsymbol{\eta}, \sigma^2\mathbf{I})$ – y_i independent with mean μ_i and shared variance σ^2 – then the C_p formula of (Mallows, 1973) provides $\mathrm{MSE}_S + 2s\sigma^2/n$ 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 (2.1) 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.[5] 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 (faster diminishing-bias) with large n or small σ.

- Plausibility of the restricted eigenvale 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. For example, the next sub-section suggests using large γ so long as λ is big enough to avoid false discovery. 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 Buhlmann and van de Geer (2011) have extensive discussion and examples.

[5] We work with C_p here, rather than AIC or $AICc$, since C_p is conveniently defined the scale of squared errors.

2.2.2 False Discovery Control

A common goal in high-dimensional estimation is support recovery – having the set $\{j : \hat{\beta}_j \neq 0\} = \{j : \beta_j \neq 0\}$ for some 'true' $\boldsymbol{\beta}$. For standard lasso estimated $\hat{\boldsymbol{\beta}}$, many authors have shown (e.g., Buhlmann and van de Geer, 2011; Zou, 2006) that to get exact support recovery asymptotically or with high probability requires an *irrepresentability condition* which limits the size of least-squares projections from 'true support' onto spurious covariates.

DEFINITION 2.2. *The $(\theta, S, \mathbf{v})$-irrepresentable condition for $\theta \in [0, 1]$ and $\mathbf{v} \in \mathbb{R}^s$ holds that,*

$$|\boldsymbol{x}_j' \mathbf{X}_S (\mathbf{X}_S' \mathbf{X}_S)^{-1} \mathbf{v}| \leq \theta \ \ \forall \ j \notin S \tag{7}$$

This is often presented with $\mathbf{v} = \mathbf{1}$.[6] It can be a strict design restriction; for example, Buhlmann and van de Geer (2011) show a single variable that is highly correlated with many columns of $\mathbf{X}_S$ leading to failure. Much of the literature on concave penalization has focused on achieving support recovery *without* such conditions; see, e.g., Fan et al. (2014) for a recent overview. Our results will require irrepresentable conditions with $\mathbf{v} = \boldsymbol{\omega}_S$, which becomes less restrictive as one is able to shrink weights ω_j for $j \in S$. See the remarks for more discussion.

Our comparison of interest is between $\hat{S} = \{j : \hat{\beta}_j \neq 0\}$, for $\hat{\boldsymbol{\beta}}$ from weighted-L_1 penalized estimation, and $S = \{j : \beta_j^\nu \neq 0\}$ for $\boldsymbol{\beta}^\nu$ the L_0 penalized estimator from Theorem 2.1. Whether looking to an L_0 oracle or a sparse truth, our experience is that exact support recovery does not occur in practice (e.g., see the simulation in Section 5.1). Thus, we instead focus on ability of the weighted-lasso to minimize *false discoveries*: $\hat{\beta}_j \neq 0$ when $\beta_j^\nu = 0$.

THEOREM 2.2. *Consider the setting of Theorem 2.1. If $\omega_{S^c}^{\min} = 1$ and $\lambda > \sqrt{2\nu}$ then*

$$\|\mathbf{X}_{S^c}' \mathbf{X}_S (\mathbf{X}_S' \mathbf{X}_S)^{-1} \boldsymbol{\omega}_S\|_\infty \leq 1 - \frac{\sqrt{2\nu}}{\lambda_t} \ \ \Rightarrow \ \ \hat{S} \cap S^c = \varnothing. \tag{8}$$

The result follows directly from the sign recovery lemma in Appendix B.

Remarks

- From Theorem 7.4 in Buhlmann and van de Geer (2011), the irrepresentability condi-

[6] Wainwright (2009) shows that (7) with $\theta = 1$, $\mathbf{v} = \mathbf{1}$ is necessary for lasso sign recovery in the *noiseless* setting.

tion holds with $|\boldsymbol{x}_j'\mathbf{X}_S(\mathbf{X}_S'\mathbf{X}_S)^{-1}\boldsymbol{\omega}_S| \leq \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}}\theta_{\text{adap}}(S)$ where $\theta_{\text{adap}}(S)$ is their 'adaptive restricted regression' coefficient. Of interest here, they show that $\theta_{\text{adap}}(S) \leq \sqrt{s}/\Lambda_{\min}(S)$ where $\Lambda_{\min}(S)$ is the minimum eigenvalue of $\mathbf{X}_S'\mathbf{X}_S/n$. Thus, (i) can be replaced by the restriction $\Lambda_{\min}(S) \geq \|\boldsymbol{\omega}_S\|(1-\sqrt{2\nu}/(\omega^{\min}_{S^c}\lambda))^{-1} = \sqrt{s}L$, with L from Theorem 2.1, and small values for L appear key in both predictive performance and support recovery.

- Without irrepresentability, limits on false discovery are more pessimistic. Convergence conditions imply that for $j \in S^c \cap \hat{S}$ we have $n\lambda\omega_j = |\boldsymbol{x}_j'(\mathbf{X}\hat{\boldsymbol{\beta}} - \mathbf{y})| \leq |\boldsymbol{x}_j'\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu)| + |\boldsymbol{x}_j'\mathbf{e}^S| \leq n\left(2\|\boldsymbol{\omega}_S\|/\phi(L,S) + \sqrt{2\nu}/\lambda\right)\ \forall\, j$. Dividing by $n\lambda\omega_j$ and counting yields

$$|S^c \cap \hat{S}| \leq \left|\frac{1}{\boldsymbol{\omega}_{S^c\cap\hat{S}}}\right|\left(\frac{2\|\omega_S\|}{\phi(L,S)} + \frac{\sqrt{2\nu}}{\lambda}\right) \tag{9}$$

Without the ability to make ω_j very big for $j \in S^c$ (e.g., as in a thresholding procedure like that of Zhou 2009), the result in (9) has little to say about false discovery control.

3 POSE and the gamma lasso

Our path of one-step estimators (POSE), in Algorithm 1, uses solutions along the sequence of decreasing penalty sizes, λ^t, as the basis for LLA weights at the next path step. In this, we are assuming a penalty specification such that $\lim_{b\to 0} c'(|b|) = 1$ and that the cost function is differentiable for $b \neq 0$. This yields a path of one-step LLA penalized coefficient estimates.

Algorithm 1 POSE

Initialize $\hat{\boldsymbol{\beta}}^0 = \mathbf{0}$, so that $\hat{S}_0 = \varnothing$.

Set $\lambda^1 > 0$ with step size $0 < \delta < 1$.

for $t = 1 \ldots T$:

$$\omega_j^t = \begin{cases} c'(|\hat{\beta}_j^{t-1}|) \text{ for } j \in \hat{S}_t \\ 1 \text{ for } j \in \hat{S}_t^c \end{cases} \tag{10}$$

$$\left[\hat{\alpha}, \hat{\boldsymbol{\beta}}\right]^t = \underset{\alpha,\beta_j\in\mathbb{R}}{\operatorname{argmin}}\ \ l(\alpha,\boldsymbol{\beta}) + n\sum_j \lambda^t\omega_j^t|\beta_j| \tag{11}$$

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

From an engineering standpoint, POSE has the same appeal as any successful path algo-

rithm: if the estimates change little from iteration t to $t+1$, then you will be able to quickly solve for a large set of candidate specifications. Following the discussion of Section 2.1, such algorithms are a natural match with one-step estimation: OSE relies upon inputs being close to the optimal solution, which is precisely the setting where path algorithms are most efficient. More rigorously, Theorem 2.1 applied to POSE yields $\hat{S}_{t-1} \cap S^c = \varnothing \Rightarrow \omega^{t\,\min}_{S^c} = 1$. Thus so long as λ is large enough, Section 2.2 demonstrates that fast diminishing ω_j will help control false discovery and improve prediction. Of course, the moment $\hat{S}_t \cap S^c \neq \varnothing$, diminishing-bias allows spurious covariates to enter with little shrinkage and can move the fit arbitrarily far away from L_0-optimality – that is, with λ too small the diminishing bias hurts your ability to estimate and predict. This is why it is essential to have a path of candidate λ^t to choose amongst.

3.1 The gamma lasso

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

$$c(\beta_j) = \log(1 + \gamma|\beta_j|), \tag{12}$$

where $\gamma > 0$. This penalty is concave with curvature $-1/(\gamma^{-1} + |\beta_j|)^2$ and it spans the range from L_0 ($\gamma \to \infty$) to L_1 ($\gamma \to 0$) costs (see Figure 2). It 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 (2013b) and Armagan et al. (2013).

GL – POSE under the log penalty – leads to line (10) being replaced by

$$\omega_j^t = \left(1 + \gamma|\hat{\beta}_j^{t-1}|\right)^{-1} \quad j = 1 \dots p \tag{13}$$

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 and, at the extreme, $\gamma = \infty$ yields a subset selection routine where a coefficient is unpenalized in all segments after it first becomes nonzero. Figure 3 shows solutions in a simple problem.

Each gamma lasso path segment is solved through coordinate descent, as detailed in Appendix D. The algorithm is implemented in `c` as part of the `gamlr` package for R. The software

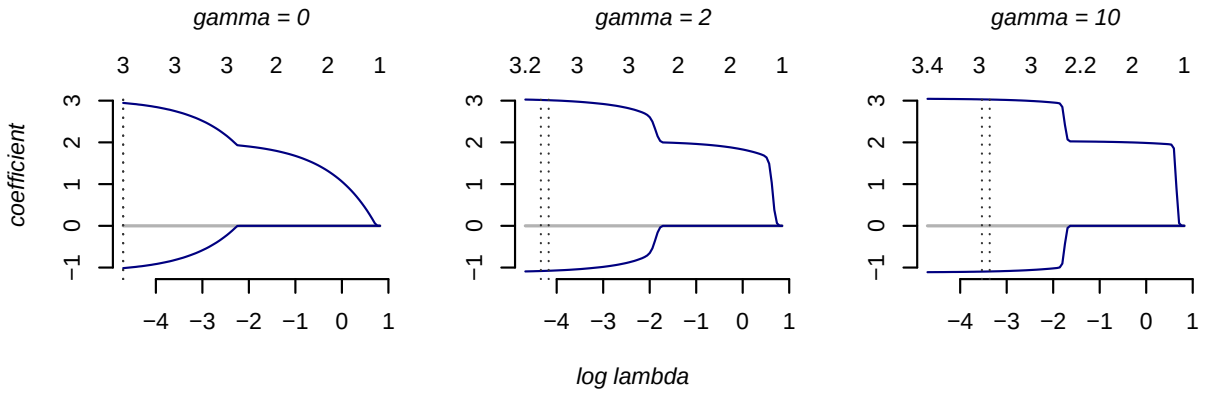

Figure 3: Gamma lasso estimation on $n = 10^3$ observations of $y_i = 4 + 3x_{1i} - x_{2i} + \varepsilon_i$, where $\varepsilon_i \overset{ind}{\sim} \mathrm{N}(0, 1)$ and $\{x_{1i}, x_{2i}, x_{3i}\}$ are marginally standard normal with correlation of 0.9 between covariates (x_{3i} is spurious). The penalty path has $T = 100$ segments, $\lambda^1 = n^{-1} |\sum_i x_{1i} y_i|$, 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).

has detailed documentation and versioned source code is at `github.com/mataddy/gamlr`. 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.

3.2 Bayesian motivation

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

$$\beta_j \sim \mathrm{La}(\tau_j) = \frac{\tau_j}{2} \exp\left[-\tau_j |\beta_j|\right]. \tag{14}$$

Typically, scale parameters $\tau_1 = \ldots = \tau_p$ are set as a single shared value, say $n\lambda/\phi$ where ϕ is the exponential family dispersion (e.g. Gaussian variance σ^2 or 1 for the binomial). Posterior maximization under the prior in (14) is then lasso estimation (e.g., Park and Casella, 2008).

Instead of working from 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 $\mathrm{var}(\tau_j) = s\gamma^2$. Then the *joint* prior for both coefficient and scale is

$$\pi(\beta_j, \tau_j) = \mathrm{La}(\beta_j;\ \tau_j)\,\mathrm{Ga}\left(\tau_j;\ s, \gamma^{-1}\right) = \frac{1}{2\Gamma(s)} \left(\frac{\tau_j}{\gamma}\right)^s \exp\left[-\tau_j(\gamma^{-1} + |\beta_j|)\right]. \tag{15}$$

The gamma hyperprior is conjugate here, implying a $\mathrm{Ga}\left(s + 1,\ 1/\gamma + |\beta_j|\right)$ posterior for $\tau_j \mid$

β_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 (15), where we've suppressed α for simplicity. By taking negative logs and removing constants, this is equivalent to solving

$$\min_{\beta_j \in \mathbb{R},\ \tau_j \in \mathbb{R}^+} \ \phi^{-1} l(\boldsymbol{\beta}) + \sum_j \left[\tau_j(\gamma^{-1} + |\beta_j|) - s\log(\tau_j)\right]. \tag{16}$$

It is straightforward to show that (16) is equivalent to the log-penalized objective

$$\min_{\beta_j \in \mathbb{R}} \ \phi^{-1} l(\boldsymbol{\beta}) + \sum_j s\log(1+\gamma|\beta_j|) \tag{17}$$

PROPOSITION 3.1. *$\hat{\boldsymbol{\beta}}$ solves (17) if and only if it is also in the solution to (16).*

Proof. The conditional posterior mode for each τ_j given β_j is $\tau(\beta_j) = \gamma s/(1+\gamma|\beta_j|)$. Any joint solution $[\hat{\boldsymbol{\beta}}, \hat{\boldsymbol{\tau}}]$ for (16) thus consists of $\hat{\tau}_j = \tau(\hat{\beta}_j)$; otherwise, it is always possible to decrease the objective by replacing $\hat{\tau}_j$. Setting each $\tau_j = \tau(\beta_j)$ in (16) and removing constant terms yields (17). Moreover, the solution to (16) solves (17): otherwise, there would need to be a point on the profile slice of (16) defined by $\tau_j = \tau(\hat{\beta}_j)$ that is lower than its minimum. □

For a Bayesian it is odd to be solving for $\boldsymbol{\tau}$ rather than marginalizing over its uncertainty. However, recognizing the form of a gamma density in (15), $\pi(\beta_j, \tau_j)$ integrates over τ_j to yield the marginal prior $\pi(\beta_j) = 0.5s\,(1+\gamma|\beta_j|)^{-(s+1)}$. This is the generalized double Pareto density, as in Armagan et al. (2013). Since $-\log\pi(\beta_j) \propto (s+1)\log(1+\gamma|\beta_j|)$, the *profile* MAP solution to (16) is also the *marginal* MAP for $\boldsymbol{\beta}$ under $\mathrm{Ga}(s-1, 1/\gamma)$ priors on each τ_j.

4 Speed, stability, and selection

As we've mentioned repeatedly, the lasso, POSE, GL and related techniques from sparse regularization do not actually *do* model selection; rather, the paths of estimators corresponding to different levels of penalization provide a set of candidates from which one must choose. It is essential that these paths are quick to obtain and are good platforms for model selection.

The lasso has become hugely popular because its path solutions have both of these properties. Diminishing bias extensions, such as GL, can provide improved prediction and infor-

mation compression over the lasso in many settings. However, when the bias diminishes too quickly (e.g., for γ too large), the paths become *instable*. For example, consider exact (not GL) solutions under the log penalty as plotted in Figure 2: the minimizer of the solid line of the right panel will jump from 0 to 3.5 depending upon small jitter either to data B or cost γs. Such instability costs us in two ways.[7]

- Computation becomes expensive: the coefficients change quickly, or even jump, along the path and are no longer good hot-starts for the next segment.
- Model selection becomes difficult: estimates are sensitive to small amount of data jitter, so that the variability explodes for any choice based upon these estimates.

This section reviews balance between the benefits of diminishing-bias and the dangers of instability, working through concrete guidelines for application of the GL algorithm.

4.1 Stability

A strong form of stability comes from convexity of the penalized objective in (1). This requires that the minimum eigenvalue of $\mathbf{H}(\boldsymbol{\beta})$, the Hessian matrix of second derivatives of $l(\boldsymbol{\beta})$, is greater than $|c''(\beta_j)|\ \forall j$. For penalized least-squares under log costs, this amounts to requiring that the minimum eigenvalue of $\mathbf{H} = \mathbf{X}'\mathbf{X}$ is greater than $\lambda\gamma^2$.[8] In the simple *standardized orthogonal covariate* case, this has an easy interpretation in the context of our Bayesian model from Section 3.2: for Gaussian regression, $h_j = \sum_i x_{ij}^2 = n$ and the objective is convex if prior variance on each τ_j is less than the number of observations. For logistic regression you need $\mathrm{var}(\tau_j) < n/4$, since $\mathbf{H}$ now depends upon the coefficient values.

In real examples, however, we cannot rely upon objective convexity. A more useful definition of stability requires continuity of the implied coefficient function, $\hat{\beta}(\mathbf{y})$, in an imagined univariate regression problem (or for orthogonal covariates). This is one of the key requirements of concave penalties listed by Fan and Li (2001). Many popular concave cost functions, such as the SCAD and MCP, have been engineered to have this continuity property. Conveniently, Zou and Li (2008) show that OSE LLA solutions have this property even if the target

[7] For a classic article on the issues of model instability, see Breiman (1996).

[8] If ν is an eigenvalue of $\mathbf{H}$, then $(\mathbf{H} - \nu\mathbf{I})\mathbf{v} = 0$ for some nonzero $\mathbf{v}$; the negative log posterior Hessian at zero is $\mathbf{H} - \lambda\gamma^2\mathbf{I}$ and $(\mathbf{H} - \lambda\gamma^2\mathbf{I} + s\gamma^2\mathbf{I} - \nu\mathbf{I})\mathbf{v} = 0$ so that $\nu - s\gamma^2$ is an eigenvalue of the minimization objective.

objective does not. For example, even though the log penalty *does not* generally lead to continuous thresholding, their result implies that the GL solutions are continuous for $\gamma < \infty$.

A theoretically richer form of stability is Lipschitz continuity of the implied prediction function, $\hat{\boldsymbol{y}} = \mathbf{X}\hat{\boldsymbol{\beta}}(\mathbf{y})$, which requires that $\|\hat{\mathbf{y}}(\mathbf{y}_1) - \hat{\mathbf{y}}(\mathbf{y}_2)| \leq L\|\mathbf{y}_1 - \mathbf{y}_2\|$ for some finite constant L on all possible $\mathbf{y}_1, \mathbf{y}_2$. Zou et al. (2007) establish Lipschitz continuity for L_1 estimated predictors as part of their derivation of a degrees-of-freedom estimator. Thus, conditional upon values for the coefficient-specific weights, POSE and GL are trivially Lipschitz continuous. Unconditionally, we do not believe that the paths have this guarantee. However, we'll see in the next section that a heuristic degrees-of-freedom estimator that takes such stability for granted performs well as the basis for model selection.

Finally, the basic and most important type of stability is practical path continuity: by this, we mean that solutions change slowly enough along the path so that computational costs are kept within budget. A regularization path can be built from a continuous thresholding function, or perhaps even be Lipschitz stable, but none of that matters if it takes too long to fit. For example, Figure 4 shows timings growing rapidly with large γ for the hockey data of Section 5.2, even though all of these specifications are theoretically stable by some criteria.

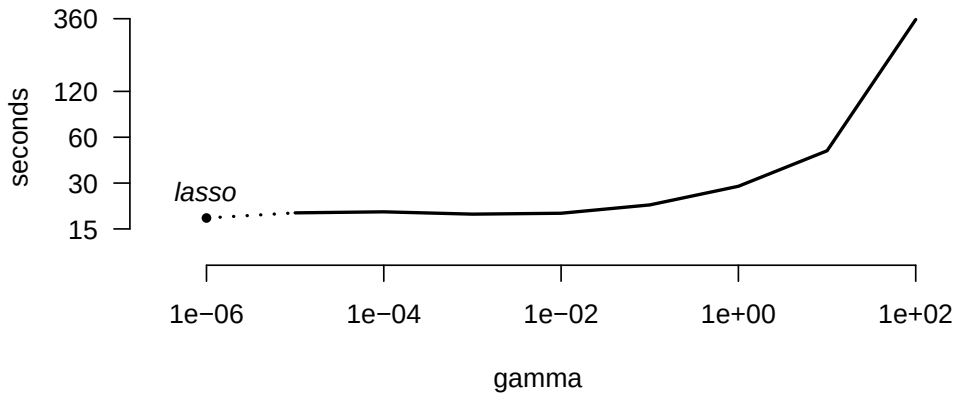

Figure 4: Timings for the hockey data path fits of Section 5.2 on a length-100 grid with $\lambda^{100} = 0.01\lambda^1$.

4.2 Selection

We would like to choose a model that performs well in predicting new data. 'Good prediction' can be measured in a variety of ways. A common and coherent framework is to consider minimizing Kullback-Leibler (KL) divergence. Say $g(\mathbf{y})$ is the true data generating process, and

$f(\mathbf{y}; \boldsymbol{\eta}, \phi)$ is the parametric density under study, which we suppose here is a natural exponential family with $\mathbb{E}[\mathbf{y}] = \boldsymbol{\eta}$ and dispersion ϕ. Then we wish to minimize

$$\mathrm{KL}(\boldsymbol{\eta}, \phi) = \mathbb{E}_g \log g(\mathbf{y}) - \mathbb{E}_g \log f(\mathbf{y}; \boldsymbol{\eta}, \phi), \tag{18}$$

the expected difference between log true density and our parametric approximation. Since $\mathbb{E}_g \log g(\mathbf{y})$ is constant, this leads one to minimize $Q(\boldsymbol{\eta}, \phi) = -\mathbb{E}_g \log f(\mathbf{y}; \boldsymbol{\eta}, \phi)$, the expected negative log likelihood. There is no requirement that g is a member of the family defined by f.

If parameters are to be estimated as $[\boldsymbol{\eta}_{\mathbf{y}}, \phi_{\mathbf{y}}]$, functions of random sample $\mathbf{y} \sim g$, then $Q(\boldsymbol{\eta}_{\mathbf{y}}, \phi_{\mathbf{y}})$ is itself a random variable and one chooses estimators to minimize its expectation. *Crucially, we imagine a double-sample expectation*, where the minimization objective is

$$\mathbb{E}_{\mathbf{y}|g} \mathbb{E}_{\tilde{\mathbf{y}}|g} \log f(\tilde{\mathbf{y}}; \boldsymbol{\eta}_{\mathbf{y}}, \phi_{\mathbf{y}}). \tag{19}$$

The notation here indicates that inner and outer expectations are based on two *independent* random samples from g: $\mathbf{y}$ for training, upon which $\boldsymbol{\eta}_{\mathbf{y}}, \phi_{\mathbf{y}}$ are calculated, and $\tilde{\mathbf{y}}$ for validation.

The remainder of this section outlines two strategies for minimization of (19). The first, cross-validation, performs a Monte Carlo experiment that mimics double expectation. The second set of methods, information criteria, attempt to approximate (19) analytically via the observed likelihood and a complexity penalty that depends upon the degrees of freedom.

4.2.1 Cross validation

K-fold cross-validation (CV; see Efron, 2004, for an overview) estimates the double expectation in (19) through a simple experiment: split the sample $\mathcal{S} = \{\mathbf{x}_i, y_i\}_{i=1}^n$ into K disjoint subsets (folds) $\mathcal{S}_k$, and K times fit a regularization path given $\mathcal{S} \setminus \mathcal{S}_k$ and use this to predict on $\mathcal{S}_k$. This yields K realizations of 'out-of-sample' (OOS) deviance, and you averaged across folds to get an estimate for (19) along the along the regularization path. One then selects λ according to some criterion and re-fits the model on the entire dataset under this penalty. The usual rules are either CV.min, choose the λ with minimum average OOS error, or CV.1se, choose the largest λ with mean OOS error no more than 1 standard error away from the minimum.

CV is super useful, and some form of it features in most applications, but it does have some

weaknesses. If a single fit is expensive then doing it K times will be impractical. More subtly, truly Big data are distributed: they are too large to store on a single machine. Algorithms can be designed to work in parallel on subsets (e.g., Taddy, 2013a), but a bottleneck results if you need to communicate across machines for OOS experimentation. Finally, large organizations may wish to avoid the Monte Carlo error of CV – they'll want the same answer every time. Thus it is useful to have an analytic estimator for (19) that requires only a single model fit.

4.2.2 Information Criteria: AIC and AICc

Information criteria (IC) are analytic approximations to metrics like (19).[9] They take the form

$$-2\log f(\mathbf{y};\boldsymbol{\eta}_\mathbf{y},\phi_\mathbf{y}) + c(df) \tag{20}$$

where $c(df)$ is cost of the *degrees-of-freedom* used in $\boldsymbol{\eta}_\mathbf{y}$ – e.g., for $\mathbf{y} \sim (\boldsymbol{\eta}, \sigma^2\mathbf{I})$, Efron et al. (2004) defines $df = \sigma^{-2}\sum_i \mathrm{cov}(\eta_{\mathbf{y}i}, y_i)$.

The commonly applied AIC uses $c(df) = 2df$ and is derived as an asymptotic approximation to (19) in Akaike (1973). However, our experience (e.g., see Section 5.1) is that AIC selects over-fit models when applied to high dimensional problems. This is because the approximation upon which it is based is valid only for large n/df (Burnham and Anderson, 2002). A 'corrected' AICc for finite n/df is derived in Hurvich and Tsai (1989), and Flynn et al. (2013) study its application to penalized deviance estimation.

$$\text{AICc:} \quad -2\log f(\mathbf{y};\boldsymbol{\eta}_\mathbf{y},\phi_\mathbf{y}) + 2df\frac{n}{n-df-1}. \tag{21}$$

The correction multiplier is $n/(n-df-1)$, and AICc approaches standard AIC if $n \gg df$. See Appendix C for a simple derivation of the AICc in linear models, Claeskens and Hjort (2008) for a more traditional derivation and full theoretical review of information criteria, and refer to Flynn et al. (2013) for results extending to generalized linear models.

In order to make use of the AICc or AIC, we need to have a value for the number of degrees-of-freedom. In an unpenalized linear model, df is just the number of coefficients. For

[9] Not all IC target (19). For example, the 'Bayesian' BIC, with $c(df) = \log(n)df$ (Schwarz, 1978), is derived (Kass and Raftery, 1995) as Laplace approximation to the negative log of the *marginal likelihood*. We include the BIC as a comparator to AIC and AICc in our examples.

some penalization schemes, such as the original least-squares lasso (Zou et al., 2007), unbiased estimates for df are available analytically. More generally, it is common to rely upon heuristic arguments (e.g., even for lasso penalized logistic regression we are unaware of theoretically unbiased estimators). A heuristic is what we'll propose here.

For prediction rules that are suitably stable (i.e., Lipschitz; see Zou et al., 2007), the SURE framework of Stein (1981) applies and we get the relatively easy-to-work-with expression $df = \mathbb{E}\left[\sum_i \partial \eta_{\mathbf{y}i}/\partial y_i\right]$. 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 $\eta_{\mathbf{y}} = x(\varsigma/h)(|g| - \tau)_+$ with derivative $\partial \eta_{\mathbf{y}i}/\partial y = x_i^2/h \mathbb{1}_{[|g|<\tau]}$, 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, as an estimator (e.g., Zou et al., 2007) one can evaluate this expression at observed gradients.

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 back to the Bayesian model of Section 3.2, each τ_j is IID $\mathrm{Ga}(s, 1/\gamma)$ in the prior, leading to the *gamma lasso estimator for degrees of freedom*[10]

$$df^t = \sum_j \mathrm{Ga}(|g_j|;\ n\lambda^t/(\gamma\phi), 1/\gamma), \tag{22}$$

where $\mathrm{Ga}(\ \cdot\ ;\ \mathrm{shape}, 1/\mathrm{scale})$ is the Gamma distribution function and g_j is an estimate of the j^{th} coefficient gradient evaluated at $\hat{\beta}_j = 0$. For fully orthogonal covariates, g_j is available as 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 $df = \hat{p}^t$ for $\gamma = 0$.

The df estimator in (22) seems intuitively reasonable; for example, $\lim_{\gamma\to 0} df^t = \hat{p}^t$ and $\lim_{\gamma\to\infty} df^t = p$. However, the derivation is purely heuristic and we rely on empirical results to justify its application as part of an AICc criteria. When used as an input to AICc, as in the next section, it allows those criteria to perform as well or better than nonparametric cross-validation.

[10]The number of unpenalized coefficients (e.g., 1 for α) is always added to this to get total df.

5 Examples

5.1 Simulation

This section will analyze data simulated from the following $p = 1000$ dimensional regression.

$$y \sim \mathrm{N}\left(\mathbf{x}'\boldsymbol{\beta}, \sigma^2\right) \text{ where } \mathbf{x} = \mathbf{u} * \mathbf{z}, \ \mathbf{u} \sim \mathrm{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right), \ z_j \overset{ind}{\sim} \mathrm{Bin}(0.5), \ \beta_j = \frac{1}{j}\exp\left(-\frac{j}{50}\right). \tag{23}$$

Each simulation draws $n = 1000$ means $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$, and two independent response samples $\mathbf{y}, \tilde{\mathbf{y}} \sim \mathrm{N}(\boldsymbol{\eta}, \sigma^2\mathbf{I})$. Residual variance σ^2 and covariate correlation $\boldsymbol{\Sigma}$ are adjusted across runs. In the first case, we define σ^2 through *signal-to-noise* ratios $\mathrm{sd}(\boldsymbol{\eta})/\sigma$ of $1/2$, 1, and 2. In the latter case, multicollinearity is parametrized via $\Sigma_{jk} = \rho^{|j-k|}$, and we consider $\rho = 0, 0.5,$ and 0.9.

The regression in (23) is obviously *dense*: true coefficients are all nonzero. However, they decay in magnitude along the index j and it will be useless to estimate many of them in a $p = n$ regression. Our sparse oracle comparator is the C_p optimal L_0 penalized solution

$$\boldsymbol{\beta}^\star = \underset{\boldsymbol{\beta}}{\mathrm{argmin}} \left\{ \|\mathbf{y} - \mathbf{X}\boldsymbol{\beta}\|^2 + 2\sigma^2 \sum_j \mathbb{1}_{\{\beta_j \neq 0\}} \right\}, \tag{24}$$

which is solvable here by searching through OLS regression on $\mathbf{X}_{\{1\ldots j\}}$ for $j = 1 \ldots p$.

We consider `gamlr` runs of GL with γ of 0 (lasso), 2, and 10 and marginal AL, as well as `sparsenet`'s MCP penalized regression. Covariate penalties are standardized by $\mathrm{sd}(\boldsymbol{x}_j)$, and paths run through 100 λ values down to $\lambda^{100} = 0.01\lambda^1$. On an Intel Xeon E5-2670 core, 5-fold CV with $\mathrm{sd}(\eta)/\sigma = 1$ and $\rho = 1/2$ requires 1-2 seconds for lasso and marginal AL, 2 and 3 seconds for GL with $\gamma = 2$ and $\gamma = 10$, and 15-20 seconds for Sparsenet. [11]

Figures 5 and 6 illustrate GL paths for a single dataset, with $\mathrm{sd}(\eta)/\sigma = 1$ and $\rho = 1/2$. In Figure 5, increasing γ leads to 'larger shouldered' paths where estimates move quickly to MLE for the nonzero-coefficients. Degrees of freedom, calculated as in (22), are along the top of each plot; equal λ have higher df^t for higher γ since there is less shrinkage of $\hat{\beta}_j \neq 0$. Figure 6 shows CV and AICc error estimates. The two criteria roughly track each other, although

[11] `ncvreg` SCAD required ten minutes for a single run and is thus impractical for the applications under consideration. But, in a small study, CV.min selected SCAD performs quite well in prediction – similarly to the best CV.min methods for each data configuration – with relatively high values for both false discovery and sensitivity.

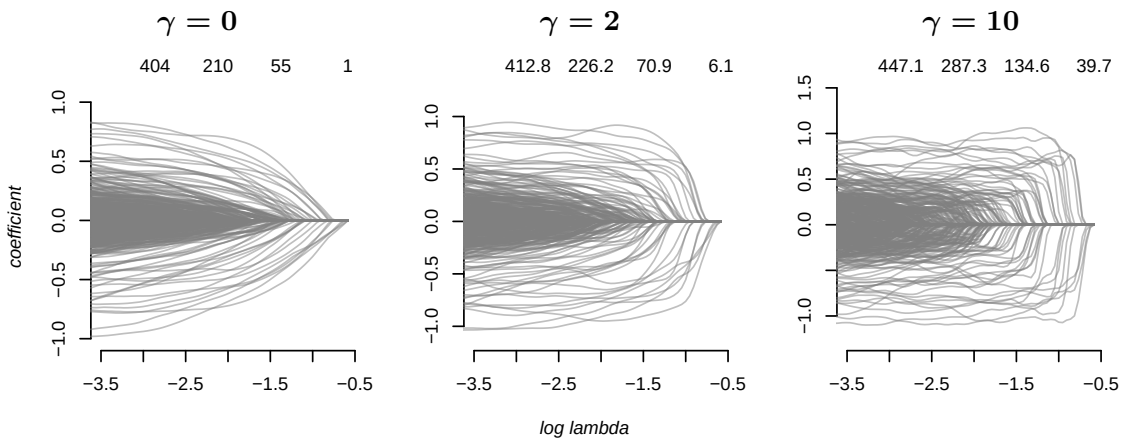

Figure 5: Regularization paths for simulation example. Degrees of freedom df^t are along the top.

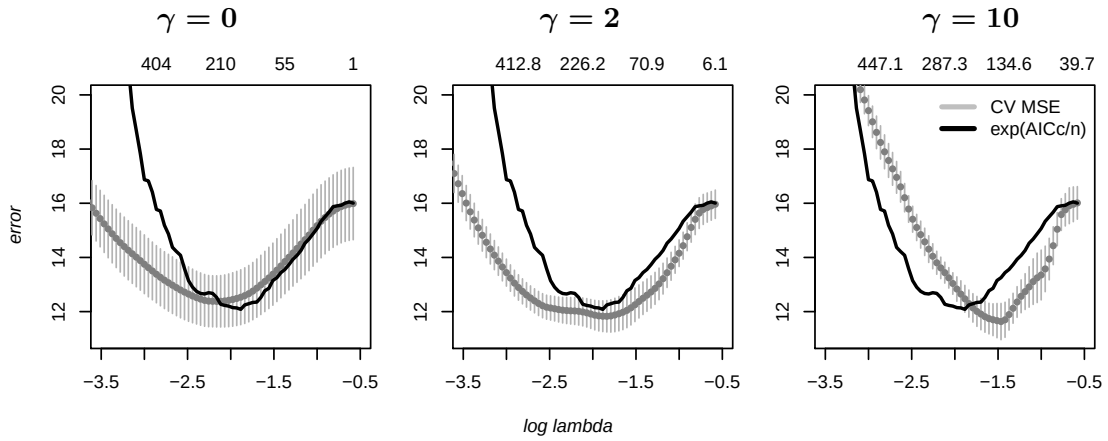

Figure 6: 5-fold CV and AICc for a simulation example. Points-and-bars show mean OOS MSE $\pm$1se.

AICc more heavily penalizes over-fit and at $\gamma = 10$ their minima do not match. Notice that as γ increases, the CV error increases more quickly after it's minimum; this indicates that the consequences of over-fit are worse under faster diminishing-bias.

Results over a set of 1000 datasets are presented in Tables 1 and 2. The first table records out-of-sample $R^2 = 1 - \mathrm{var}(\tilde{\mathbf{y}} - \boldsymbol{\eta}_{\mathbf{y}})/\mathrm{var}(\tilde{\mathbf{y}})$, while the second reports false discovery and sensitivity with respect to the L_0 oracle of (24). For $\mathrm{sd}(\eta)/\sigma$ of 1 to 2, the highest R^2 values are obtained by CV.min MCP and by $\gamma = 2$ GL with either CV.min or AICc selection. When $\mathrm{sd}(\eta)/\sigma$ drops to $1/2$, the top performer is AICc lasso. Investigation of the AICc values finds that $\gamma = 0$ was consistently favored over $\gamma = 2$ in this low signal setting, so that AICc selection on just two GL paths (for each of $\gamma = 0$ and 1) would have been enough to get the best predictor in all scenarios.[12] The very sparse $\gamma = 10$ fits do nearly as well as best except in the low signal

[12]Note also the massive improvement of AICc over AIC: the latter leads to negative R^2.

Table 1: *Predictive* R^2

	lasso	GL $\gamma = 2$	GL $\gamma = 10$	marginal AL	sparsenet MCP	
CV.1se	0.71	0.72	0.68	0.72	0.73	
CV.min	0.73	**0.74**	0.72	0.73	**0.74**	$\mathrm{sd}(\mu)/\sigma = 2$
AICc	0.72	0.73	0.73	0.73		$\rho = 0$
AIC	0.67	0.66	0.66	0.73		C_p R^2 = 0.77
BIC	0.63	0.67	0.63	0.68		
CV.1se	0.7	0.71	0.67	0.7	0.72	
CV.min	0.72	**0.73**	0.7	0.71	**0.73**	$\mathrm{sd}(\mu)/\sigma = 2$
AICc	0.71	**0.73**	0.72	0.71		$\rho = 0.5$
AIC	0.67	0.66	0.66	0.71		C_p R^2 = 0.77
BIC	0.57	0.64	0.6	0.63		
CV.1se	0.7	0.71	0.68	0.68	0.72	
CV.min	0.72	**0.73**	0.71	0.7	**0.73**	$\mathrm{sd}(\mu)/\sigma = 2$
AICc	0.71	**0.73**	0.72	0.7		$\rho = 0.9$
AIC	0.68	0.67	0.67	0.7		C_p R^2 = 0.77
BIC	0.57	0.63	0.6	0.61		
CV.1se	0.31	0.31	0.27	0.36	0.33	
CV.min	0.36	0.36	0.33	0.36	0.36	$\mathrm{sd}(\mu)/\sigma = 1$
AICc	0.35	**0.37**	0.35	0.36		$\rho = 0$
AIC	0.13	0.09	0.09	0.3		C_p R^2 = 0.44
BIC	0.15	0.22	0.04	0.28		
CV.1se	0.27	0.29	0.23	0.33	0.32	
CV.min	0.34	**0.35**	0.3	0.33	**0.35**	$\mathrm{sd}(\mu)/\sigma = 1$
AICc	0.32	**0.35**	**0.35**	0.33		$\rho = 0.5$
AIC	0.12	0.1	0.1	0.28		C_p R^2 = 0.44
BIC	0.04	0.13	0.02	0.19		
CV.1se	0.28	0.29	0.23	0.3	0.31	
CV.min	0.34	**0.35**	0.31	0.32	**0.35**	$\mathrm{sd}(\mu)/\sigma = 1$
AICc	0.33	**0.35**	**0.35**	0.32		$\rho = 0.9$
AIC	0.14	0.11	0.11	0.3		C_p R^2 = 0.44
BIC	0.03	0.07	0.01	0.13		
CV.1se	0.01	0	0	**0.06**	0	
CV.min	**0.06**	0.02	0.01	0.01	0.05	$\mathrm{sd}(\mu)/\sigma = 0.5$
AICc	**0.06**	0.02	0	0.04		$\rho = 0$
AIC	-0.44	-0.51	-0.51	-0.2		C_p R^2 = 0.13
BIC	0	0	0	0.01		
CV.1se	0	0	0	0.03	0	
CV.min	0.03	0.01	0	-0.01	0.03	$\mathrm{sd}(\mu)/\sigma = 0.5$
AICc	**0.04**	0.01	0	0.01		$\rho = 0.5$
AIC	-0.45	-0.51	-0.51	-0.21		C_p R^2 = 0.13
BIC	0	0	0	0		
CV.1se	0	0	0	0.02	0	
CV.min	0.02	0.01	0	0	0.02	$\mathrm{sd}(\mu)/\sigma = 0.5$
AICc	**0.03**	0.01	0	0.01		$\rho = 0.9$
AIC	-0.43	-0.5	-0.5	-0.17		C_p R^2 = 0.13
BIC	0	0	0	0		

Table 2: *False Discovery Rate | Sensitivity, relative to C_p model*

	lasso	GL $\gamma = 2$	GL $\gamma = 10$	marginal AL	sparsenet MCP	
CV.1se	0.49 | 0.76	0.26 | 0.67	0.07 | 0.56	0.43 | 0.67	0.15 | 0.61	
CV.min	0.65 | 0.84	0.48 | 0.77	0.20 | 0.65	0.57 | 0.74	0.40 | 0.74	$\mathrm{sd}(\mu)/\sigma = 2$
AICc	0.55 | 0.79	0.44 | 0.76	0.37 | 0.73	0.53 | 0.71		$\rho = 0$
AIC	0.84 | 0.94	0.84 | 0.93	0.84 | 0.91	0.63 | 0.77		$\bar{s}_{C_p}$ = 123.7
BIC	0.22 | 0.59	0.07 | 0.53	0.02 | 0.45	0.22 | 0.54		
CV.1se	0.58 | 0.75	0.37 | 0.67	0.12 | 0.55	0.53 | 0.64	0.14 | 0.58	
CV.min	0.69 | 0.84	0.56 | 0.77	0.27 | 0.65	0.63 | 0.71	0.34 | 0.69	$\mathrm{sd}(\mu)/\sigma = 2$
AICc	0.60 | 0.77	0.50 | 0.74	0.41 | 0.73	0.59 | 0.69		$\rho = 0.5$
AIC	0.84 | 0.94	0.84 | 0.94	0.84 | 0.92	0.67 | 0.75		$\bar{s}_{C_p}$ = 123.5
BIC	0.29 | 0.53	0.13 | 0.49	0.04 | 0.44	0.31 | 0.49		
CV.1se	0.59 | 0.76	0.43 | 0.67	0.21 | 0.56	0.54 | 0.61	0.20 | 0.57	
CV.min	0.69 | 0.84	0.57 | 0.77	0.35 | 0.66	0.63 | 0.69	0.41 | 0.69	$\mathrm{sd}(\mu)/\sigma = 2$
AICc	0.61 | 0.77	0.52 | 0.74	0.43 | 0.71	0.60 | 0.66		$\rho = 0.9$
AIC	0.84 | 0.94	0.84 | 0.93	0.83 | 0.91	0.65 | 0.71		$\bar{s}_{C_p}$ = 123.9
BIC	0.35 | 0.51	0.20 | 0.47	0.09 | 0.43	0.39 | 0.47		
CV.1se	0.43 | 0.58	0.18 | 0.44	0.07 | 0.33	0.52 | 0.62	0.26 | 0.50	
CV.min	0.66 | 0.73	0.43 | 0.60	0.19 | 0.45	0.68 | 0.73	0.57 | 0.69	$\mathrm{sd}(\mu)/\sigma = 1$
AICc	0.60 | 0.69	0.50 | 0.65	0.50 | 0.63	0.62 | 0.69		$\rho = 0$
AIC	0.90 | 0.93	0.90 | 0.92	0.90 | 0.89	0.79 | 0.82		$\bar{s}_{C_p}$ = 90.0
BIC	0.07 | 0.25	0.04 | 0.25	0.00 | 0.04	0.17 | 0.38		
CV.1se	0.52 | 0.54	0.30 | 0.43	0.12 | 0.32	0.61 | 0.58	0.23 | 0.42	
CV.min	0.70 | 0.71	0.53 | 0.60	0.27 | 0.45	0.71 | 0.69	0.47 | 0.58	$\mathrm{sd}(\mu)/\sigma = 1$
AICc	0.65 | 0.66	0.56 | 0.63	0.56 | 0.65	0.67 | 0.65		$\rho = 0.5$
AIC	0.90 | 0.94	0.90 | 0.93	0.90 | 0.91	0.81 | 0.81		$\bar{s}_{C_p}$ = 90.0
BIC	0.03 | 0.07	0.05 | 0.15	0.00 | 0.03	0.21 | 0.26		
CV.1se	0.57 | 0.54	0.40 | 0.43	0.23 | 0.30	0.62 | 0.53	0.32 | 0.40	
CV.min	0.71 | 0.72	0.57 | 0.60	0.38 | 0.45	0.70 | 0.65	0.55 | 0.58	$\mathrm{sd}(\mu)/\sigma = 1$
AICc	0.66 | 0.66	0.58 | 0.61	0.58 | 0.62	0.69 | 0.62		$\rho = 0.9$
AIC	0.90 | 0.94	0.90 | 0.92	0.90 | 0.90	0.80 | 0.77		$\bar{s}_{C_p}$ = 89.7
BIC	0.03 | 0.04	0.05 | 0.08	0.00 | 0.01	0.21 | 0.17		
CV.1se	0.07 | 0.05	0.01 | 0.01	0.01 | 0.01	0.65 | 0.45	0.07 | 0.04	
CV.min	0.61 | 0.41	0.20 | 0.13	0.07 | 0.05	0.80 | 0.63	0.59 | 0.40	$\mathrm{sd}(\mu)/\sigma = 0.5$
AICc	0.64 | 0.43	0.49 | 0.34	0.02 | 0.01	0.76 | 0.57		$\rho = 0$
AIC	0.94 | 0.92	0.94 | 0.90	0.94 | 0.90	0.90 | 0.81		$\bar{s}_{C_p}$ = 56.5
BIC	0.00 | 0.01	0.00 | 0.00	0.00 | 0.00	0.07 | 0.05		
CV.1se	0.04 | 0.01	0.01 | 0.00	0.00 | 0.00	0.70 | 0.35	0.09 | 0.01	
CV.min	0.56 | 0.25	0.20 | 0.07	0.08 | 0.03	0.82 | 0.56	0.54 | 0.24	$\mathrm{sd}(\mu)/\sigma = 0.5$
AICc	0.65 | 0.31	0.43 | 0.26	0.05 | 0.03	0.79 | 0.49		$\rho = 0.5$
AIC	0.94 | 0.93	0.94 | 0.91	0.94 | 0.91	0.91 | 0.80		$\bar{s}_{C_p}$ = 56.3
BIC	0.01 | 0.00	0.00 | 0.00	0.00 | 0.00	0.06 | 0.02		
CV.1se	0.03 | 0.01	0.01 | 0.00	0.00 | 0.00	0.68 | 0.25	0.14 | 0.01	
CV.min	0.53 | 0.19	0.23 | 0.05	0.08 | 0.02	0.82 | 0.49	0.52 | 0.17	$\mathrm{sd}(\mu)/\sigma = 0.5$
AICc	0.67 | 0.27	0.41 | 0.21	0.04 | 0.03	0.80 | 0.44		$\rho = 0.9$
AIC	0.94 | 0.93	0.94 | 0.90	0.94 | 0.90	0.90 | 0.76		$\bar{s}_{C_p}$ = 55.8
BIC	0.01 | 0.00	0.00 | 0.00	0.00 | 0.00	0.08 | 0.01		

settings, where all but lasso suffer. No algorithm ever does more than 1-2% better than lasso.

In Table 2, where the false discovery rate (FDR) $\sum_j \mathbb{1}_{\{\hat{\beta}_j \neq 0 \cap \beta_j^\star = 0\}} / \sum_j \mathbb{1}_{\{\hat{\beta}_j \neq 0\}}$ needs to be balanced against sensitivity $\sum_j \mathbb{1}_{\{\hat{\beta}_j \neq 0 \cap \beta_j^\star \neq 0\}} / \sum_j \mathbb{1}_{\{\beta_j^\star \neq 0\}}$, results are less straightforward. If false discovery is the primary concern, then you do well using GL with $\gamma = 10$; for example, under CV.1se selection when $\mathrm{sd}(\eta)/\sigma = 1$ and $\rho = 1/2$, $\gamma = 10$ GL has FDR of 0.12 against lasso's 0.52 and MCP's 0.23. But this does come at the expense of a drop in sensitivity, to 0.32 from lasso's 0.54 and MCP's 0.42. Across all routines, CV.1se selection appears to do the best job of controlling FDR without too dramatically under-fitting. In any case, it appears that you should use small γ when focused on prediction and large γ when FDR is a primary concern.

5.2 Hockey players

This section attempts to quantify the performance of hockey players. It extends analysis in Gramacy et al. (2013). The current version includes data about who was on the ice for every goal in the National Hockey League (NHL) back to the 2002-2003 season, including playoffs. The data are in the `gamlr` package; there are 69449 goals and 2439 players.

The logistic regression model of player contribution is, for goal i in season s with away team a and home team h,

$$\text{logit}\left[\text{p(home team scored goal } i)\right] = \alpha_0 + \alpha_{sh} - \alpha_{sa} + \mathbf{u}_i'\boldsymbol{\phi} + \mathbf{x}_i'\boldsymbol{\beta}_0 + \mathbf{x}_i'\boldsymbol{\beta}_s, \qquad (25)$$

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 gamma lasso paths of $\boldsymbol{\beta}$ for the model in (25) *with $\boldsymbol{\alpha}$ and $\boldsymbol{\phi}$ left unpenalized*. In contrast to the default for most analyses, our coefficient costs are *not* scaled by covariate

	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.

standard deviation. Doing the usual standardization would have favored players with little ice time. The algorithm is run for $\log_{10} \gamma = -5 \ldots 2$, plus the $\gamma = 0$ lasso.[13]

Joint $[\gamma, \lambda]$ surfaces for AICc and BIC are in Figure 7. 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 at $\gamma = 100$, which is where timings explode in Figure 4. Ten-fold CV results are shown in Figure 8 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. We also see that AICc selection is always between the CV.min and CV.1se selctions: at $\gamma = 0$ AICc matches the CV.1se choice, while at $\gamma = 10$ it has moved right to the CV.1se selection. Our heuristic from Section 4.2 might be over-estimating df for large-γ models (especially under this very collinear design), but one would also suspect that CV estimates of minimum deviance are biased downward more dramatically for larger γ than for low-variance small-γ estimators.

[13] On this data, $\gamma = \infty$ subset selection yields perfect separation and infinite likelihood.

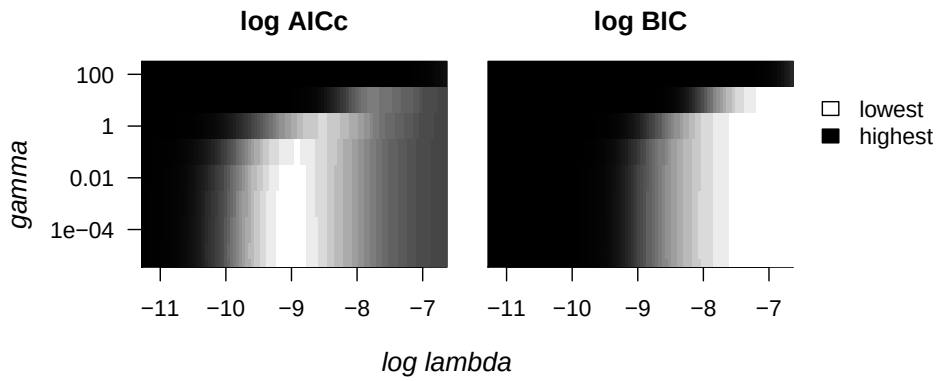

Figure 7: Hockey example AICc and BIC surfaces, rising from white to black on log scale.

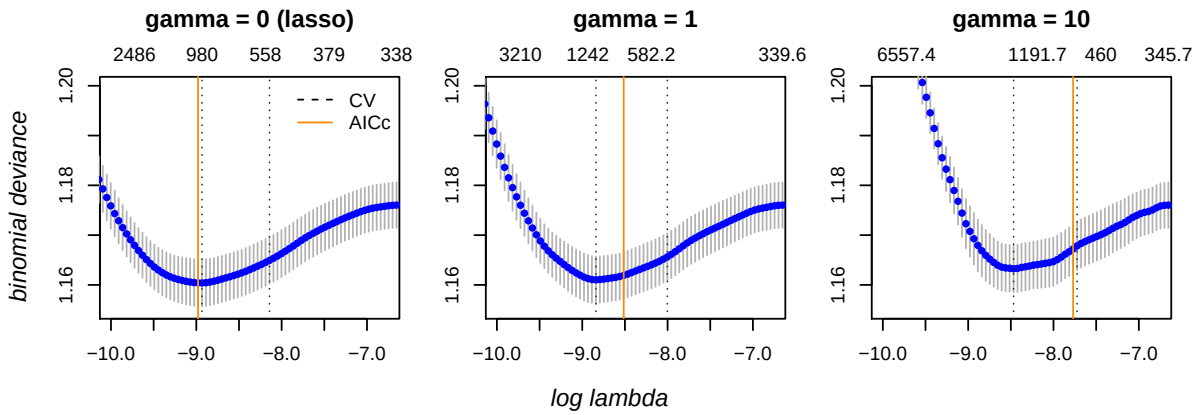

Figure 8: Hockey example 10-fold CV: mean OOS deviance $\pm$1se, with minimum-error and 1SE selection rules marked with black dotted lines, and solid orange line showing AICc selection.

The motivating application for this example was to devise a better versions 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', we first state that in absence of any other relevant information about each goal (not team, home/away, etc), the probability a goal was scored by his team given player j is on ice becomes $p_j = e^{\beta_j}/(1+e^{\beta_j})$ and our 'partial plus/minus' (PPM) is

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

where N_j is the number of goals for which he was on-ice. This measures quality and quantity of contribution, controlling for confounding information, and lives on the same scale as PM.

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, 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.

6 Discussion

Concave penalized estimation in Big data, where exact solvers are too computationally expensive, reduces largely to weighted-L_1 penalization. This review has covered a number of topics that we think relevant for such schemes. Apart from the simulation study, we have not provided extensive comparison of the many available weighting mechanisms. However, we feel that path adaptation is an intuitively reasonable source of weights. In any case, POSE has the advantage that using a regularization path to supply penalty weights is computationally efficient. To scale for truly Big data, L_1 weights need be constructed at practically no cost on top of a standard lasso run. Beyond `gamlr` and marginal regression adaptive lasso, we have found no other software for sparse diminishing bias estimation where this standard is met.

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. *To appear in Statistica Sinica*.

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, August). 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.

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

Burnham, K. and D. Anderson (2002). *Model Selection and Multimodel Inference* (2 ed.). 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)*.

Claeskens, G. and N. L. Hjort (2008). *Model selection and model averaging*. Cambridge; New York: Cambridge University Press.

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, June). 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, H. Hofling, and R. Tibshirani (2007). Pathwise coordinate optimization. *The Annals of Applied Statistics 1*, 302–332.

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.

Goldberger, A. S. (1961, December). Stepwise least squares: Residual analysis and specification error. *Journal of the American Statistical Association 56*(296), 998.

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

Green, P. J. (1984). Iteratively reweighted least squares for maximum likelihood estimation, and some robust and resistant alternatives. *Journal of the Royal Statistical Society, Series B 46*, 149–192.

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.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*, 773–795.

Lange, K. (2010). *Numerical Analysis for Statisticians* (2nd ed.). Springer.

Lee, J., Y. Sun, and M. Saunders (2014). Proximal newton-type methods for minimizing convex objective functions in composite form. *SIAM Journal on Optimization 24*, 1420–1443.

Luenberger, D. G. and Y. Ye (2008). *Linear and Nonlinear Programming* (3rd ed.). Springer.

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. (2013a). Distributed multinomial regression. arXiv:1311.6139.

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

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

Tseng, P. (2001). Convergence of a block coordinate descent method for nondifferentiable minimization. *Journal of Optimization Theory and Applications 109*, 475–494.

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

Wainwright, M. J. (2009, May). Sharp thresholds for high-dimensional and noisy sparsity recovery using L1-constrained quadratic programming (lasso). *IEEE Transactions on Information Theory 55*(5), 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.

Wu, T. T. and K. Lange (2008). Coordinate descent algorithms for lasso penalized regression. *The Annals of Applied Statistics 2*, 1–21.

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, November). Multi-stage convex relaxation for feature selection. *Bernoulli 19*(5B), 2277–2293.

Zhou, S. (2009). Thresholding procedures for high-dimensional variable selection and statistical estimation.pdf. *Advances in Neural Information Processing Systems 22*.

Zhou, S., S. van de Geer, and P. Bhlmann (2009). Adaptive lasso for high dimensional regression and gaussian graphical modeling. *arXiv preprint arXiv:0903.2515*.

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*(4), 1509–1533.

Appendices

A Gradient, curvature, and path starts

The negative log likelihood objective in Gaussian regression is $l(\alpha, \boldsymbol{\beta}) = 0.5 \sum_i (y_i - \eta_i)^2$ with gradient $g_j(\boldsymbol{\beta}) = \partial l / \partial \beta_j = -\sum_i x_{ij}(y_i - \eta_i)$, and coordinate curvature $h_j(\boldsymbol{\beta}) = \partial^2 l / \partial \beta_j^2 = \sum_i x_{ij}^2$. In logistic regression, set $y_i = 1$ for 'success' and $y_i = 0$ for 'failure' and write $q_i = (1+\exp[-\eta_i])^{-1}$ as the probability of success. Then $l(\alpha, \boldsymbol{\beta}) = \sum_i -y_i\eta_i + \log(1+\exp[\eta_i])$, $g_j(\boldsymbol{\beta}) = \partial l / \partial \beta_j = -\sum_i x_{ij}(y_i - q_i)$, and $h_j(\boldsymbol{\beta}) = \partial^2 l / \partial \beta_j^2 = \sum_i x_{ij}^2 q_i(1 - q_i)$. In each case, it is implied that $\hat{\alpha}$ has been set to minimize $l(\alpha, \hat{\boldsymbol{\beta}})$.

For L_1 costs $c_j(|\beta_j|) = |\beta_j|$, the infimum λ such that $\hat{\boldsymbol{\beta}} = \mathbf{0}$ is available analytically as $\lambda^1 = n^{-1} \max\{|g_j(\mathbf{0})|,\ j = 1 \dots p\}$, the maximum mean absolute gradient for the null model with $\boldsymbol{\beta} = \mathbf{0}$. This formula is used to obtain our starting values for the path algorithms.

B Sign Recovery

LEMMA B.1. *Under the setting of Theorem 2.1, with* $\hat{S} = \{j : \hat{\beta}_j \neq 0\}$, *if* $\omega_{S^c}^{\min} \lambda > \sqrt{2\nu}$ *then*

$$|\boldsymbol{x}_j' \mathbf{X}_S (\mathbf{X}_S' \mathbf{X}_S)^{-1} \boldsymbol{\omega}_S| \leq 1 - \frac{\sqrt{2\nu}}{\lambda \omega_j} \ \forall\ j \ \in \ S^c \Rightarrow \hat{S} \cap S^c = \varnothing. \tag{26}$$

If in addition $|(\mathbf{X}_S' \mathbf{X}_S)^{-1} \mathbf{X}_S' \mathbf{y}|_\infty > n\lambda\, |(\mathbf{X}_S' \mathbf{X}_S)^{-1} \boldsymbol{\omega}_S|_\infty$, *then* $\operatorname{sgn}(\hat{\boldsymbol{\beta}}) = \operatorname{sgn}(\boldsymbol{\beta}^\nu)$.

Proof. The Karush-Kuhn-Tucker (KKT) conditions at weighted-L_1 minimization convergence imply that

$$\mathbf{x}_j' \mathbf{X} (\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu) + \mathbf{x}_j' \mathbf{e}^S = -n\lambda \zeta_j \ \text{ for } \ j = 1 \dots p \tag{27}$$

where $|\zeta_j| = \omega_j$ for $j \in \hat{S}$ and $|\zeta_j| \leq \omega_j$ for $j \in \hat{S}^c$. Following closely related proofs in Wainwright (2006, 2009); Zhou et al. (2009), $\hat{S} \cap S^c = \varnothing$ occurs if and only if these KKT conditions hold for projections restricted to S,

$$\mathbf{X}_S' \mathbf{X}_S (\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^\nu) + \mathbf{X}_S' \mathbf{e}^S = -n\lambda \boldsymbol{\zeta}_S \ \ \Rightarrow \ \ \hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^\nu = -n\lambda (\mathbf{X}_S' \mathbf{X}_S)^{-1} \boldsymbol{\zeta}_S. \tag{28}$$

Thus all of the spurious regressors in S^c will have $\hat{\beta}_j = 0$ if and only if

$$\boldsymbol{x}_j' \mathbf{X}_S (\hat{\boldsymbol{\beta}}_S - \boldsymbol{\beta}_S^\nu) - \boldsymbol{x}_j' \mathbf{e}^S \leq n\lambda\zeta_j \ \ \Leftarrow \ \ 1 - \frac{|x_j' \mathbf{e}^S|}{n} \geq 1 - \frac{\sqrt{2\nu}}{\lambda\omega_j} \geq |\boldsymbol{x}_j' \mathbf{X}_S (\mathbf{X}_S' \mathbf{X}_S)^{-1} \boldsymbol{\omega}_S|. \tag{29}$$

Finally, for sign recovery on $j \in S$ we need $|\beta_j^\nu| - |\beta_j^\nu - \hat{\beta}_j| > 0 \ \forall \ j \ \in \ S$, and our stated condition follows from $\boldsymbol{\beta^\nu}_S = (\mathbf{X}_S'\mathbf{X}_S)^{-1}\mathbf{X}_S'\mathbf{y}$ and $\boldsymbol{\beta^\nu}_S - \hat{\boldsymbol{\beta}}_S = n\lambda(\mathbf{X}_S'\mathbf{X}_S)^{-1}\boldsymbol{\zeta}_S$. □

C Derivation of AIC and AICc

Consider a Gaussian regression model where $\boldsymbol{\eta}_\mathbf{y}$ is an estimate for $\boldsymbol{\eta} = \mathbb{E}\mathbf{y}$ using df degrees of freedom, and set $\phi_\mathbf{y} = \sigma_\mathbf{y}^2 = \sum_i (y_i - \eta_{\mathbf{y}i})^2/n$. We'll derive

$$df \frac{n}{n - df - 1} \approx \mathbb{E}_{\mathbf{y}|g} \left[\log f(\mathbf{y}; \boldsymbol{\eta}_\mathbf{y}, \phi_\mathbf{y}) - \mathbb{E}_{\tilde{\mathbf{y}}|g} \log f(\tilde{\mathbf{y}}; \boldsymbol{\eta}_\mathbf{y}, \phi_\mathbf{y}) \right], \tag{30}$$

such that AICc's complexity penalty is the expected bias that results from taking the fitted log likelihood as an estimate for (19). First, by cancellation the inner term of (30) simplifies as

$$\log f(\mathbf{y}; \boldsymbol{\eta}_\mathbf{y}, \phi_\mathbf{y}) - \mathbb{E}_{\tilde{\mathbf{y}}|g} \log f(\tilde{\mathbf{y}}; \boldsymbol{\eta}_\mathbf{y}, \phi_\mathbf{y}) = \frac{\mathbb{E}_{\tilde{\mathbf{y}}|g} \sum_i (\tilde{y}_i - \eta_{\mathbf{y}i})^2}{2\sigma_\mathbf{y}^2} - \frac{n}{2}. \tag{31}$$

Now, assume that the *true* model is linear and that the data were generated from $\mathbf{y} \sim g(\boldsymbol{\eta}, \sigma^2\mathbf{I})$. The Mallows (1973) C_p formula holds that $n\sigma_\mathbf{y}^2 + 2\sigma^2 df$ is an unbiased estimator for expected sum of square errors $\mathbb{E}_{\tilde{\mathbf{y}}|g} \sum_i (\tilde{y}_i - \eta_{\mathbf{y}i})^2/n$, such that

$$\frac{\mathbb{E}_{\tilde{\mathbf{y}}|g} \sum_i (\tilde{y}_i - \eta_{\mathbf{y}i})^2}{2\sigma_\mathbf{y}^2} - \frac{n}{2} \ \approx \ \frac{n\sigma_\mathbf{y}^2 + 2\sigma^2 df}{2\sigma_\mathbf{y}^2} - \frac{n}{2} \ = \ df \frac{\sigma^2}{\sigma_\mathbf{y}^2}. \tag{32}$$

At this point, we see that the standard AIC approximation results from equating $\sigma^2 \approx \mathbb{E}_{\mathbf{y}|g}\sigma_\mathbf{y}^2$, so that $df\,\mathbb{E}_{\mathbf{y}|g}[\sigma^2/\sigma_\mathbf{y}^2] \approx df$. This will underpenalize complexity whenever residual variance $\sigma_\mathbf{y}^2$ tends to be smaller than the true variance σ^2 – that is, whenever the model is overfit. In contrast, AICc applies the chi-squared goodness of fit result $n\sigma_\mathbf{y}^2/\sigma^2 \sim \chi^2_{n-df-1}$ to obtain

$$\mathbb{E}_{\mathbf{y}|g} \left[\frac{\sigma^2}{\sigma_\mathbf{y}^2} df \right] = n\mathbb{E}_{\mathbf{y}|g} \left[\frac{1}{n\sigma_\mathbf{y}^2/\sigma^2} \right] df = \frac{n}{n - df - 1} df. \tag{33}$$

Multiplying by -2 and subtracting from $-2 \log f(\mathbf{y}; \boldsymbol{\eta}_\mathbf{y}, \sigma_\mathbf{y})$ yields the AICc.

D Implementation via coordinate descent

We use Coordinate descent (CD; e.g., Luenberger and Ye, 2008) to minimize (11) at each step along the path. CD is a local optimization algorithm that cycles through minimization of the conditional objective for individual parameters when the remaining parameters are fixed. Algorithms of this type have have become popular in L_1 penalized estimation since the work by Friedman et al. (2007) and Wu and Lange (2008).

Our CD routine, outlined in Algorithm 2, is a solver for penalized weighted-least squares problems as defined in equation (34) below. This applies directly in Gaussian regression, and for non-Gaussian models we follow Friedman et al. (2010) and apply CD inside an outer loop of iteratively re-weighted-least-squares (IRLS; e.g., Green, 1984). Given current parameter values $\hat{\boldsymbol{\beta}}$, the Newton-Raphson update for maximum likelihood estimation is $\boldsymbol{\beta} = \hat{\boldsymbol{\beta}} - \mathbf{H}^{-1}\mathbf{g}$, where $\mathbf{H}$ is the information matrix with elements $h_{jk} = \partial^2 l / \partial\beta_j \partial\beta_k|_{\hat{\boldsymbol{\beta}}}$ and $\mathbf{g}$ is coefficient gradient (see Appendix A). For exponential family linear models we can write $\mathbf{H} = \mathbf{X}'\mathbf{V}\mathbf{X}$ and $\mathbf{g} = \mathbf{X}'\mathbf{V}(\mathbf{z} - \hat{\boldsymbol{\eta}})$, where $\mathbf{V} = \mathrm{diag}(\mathbf{v})$, $\mathbf{v} = [v_1 \dots v_n]$ are 'weights', $\mathbf{z} = [z_1 \dots z_n]$ are transformed 'response', and $\hat{\eta}_i = \hat{\alpha} + \mathbf{x}_i\hat{\boldsymbol{\beta}}$. In Gaussian regression, $v_i = 1$, $z_i = \hat{\eta}_i - y_i$, and the update is an exact solution. For binomial regression, $v_i = q_i(1 - q_i)$ and $z_i = \hat{\eta}_i - (y_i - q_i)/v_i$, where $q_i = (1 + \exp[-\hat{\eta}_i])^{-1}$ is the estimated probability of success.

This yields $\boldsymbol{\beta} = (\mathbf{X}'\mathbf{V}\mathbf{X})^{-1}\mathbf{X}'\mathbf{V}\mathbf{z}$, such that the Newton update solves a weighted-least-squares problem. Adding L_1 costs, the minimization objective from (11) becomes

$$\underset{\alpha,\beta_1 \dots \beta_p \in \mathbb{R}}{\operatorname{argmin}} \sum_i \frac{v_i}{2}(\alpha + \mathbf{x}_i'\boldsymbol{\beta} - z_i)^2 + n\sum_j \omega_j \lambda |\beta_j|. \tag{34}$$

Our solver iterates between CD on (34) and, for non-Gaussian models, updates to $\mathbf{v}$ and $\mathbf{z}$. Each t^{th} segment IRLS routine initializes $[\hat{\alpha}, \hat{\boldsymbol{\beta}}]$ at solutions for λ^{t-1}, or at $[\hat{\alpha}, \mathbf{0}]$ for $t = 1$. In the `gamlr` implementation, a full pass update of all parameters is done only at the first CD iteration; otherwise coordinates with currently inactive (zero) $\hat{\beta}_j$ are not updated. Once the descent converges for this *active set*, IRLS $\mathbf{v}$ and $\mathbf{z}$ are updated and we begin a new CD loop with a full pass update. The routine stops when maximum squared change in β_j scaled by its information over one of these full pass updates is less than some tolerance threshold, `thresh`. The default in `gamlr` uses a relative tolerance of 10^{-7} times null model deviance.

Algorithm 2 Coordinate descent

Set $\texttt{vh}_\texttt{j} = \sum_i v_i(x_{ij} - \bar{x}_j)^2$ and $\texttt{vx}_\texttt{j} = \sum_i v_i x_{ij}$ for $j = 1 \dots p$.
while $\max_{j=1\dots p} \texttt{vh}_\texttt{j}\Delta_j^2 > \texttt{thresh}$:
    for j=1…p:
        set $\texttt{vg}_\texttt{j} = -\sum_i x_{ij} v_i (z_i - \hat{\eta}_i)$ and $\texttt{ghb} = \texttt{vg}_\texttt{j} - \texttt{vh}_\texttt{j}\hat{\beta}_j$
        if $|\texttt{ghb}| < n\lambda^t\omega_j^t$: $\Delta_j = -\hat{\beta}_j$
        else: $\Delta_j = -(\texttt{vg}_\texttt{j} - \text{sign}(\texttt{ghb})n\lambda^t\omega_j^t)/\texttt{vh}_\texttt{j}$.
        update $\hat{\beta}_j \stackrel{+}{=} \Delta_j$, $\hat{\alpha} \stackrel{+}{=} -\texttt{vx}_\texttt{j}\Delta_j$, and $\hat{\boldsymbol{\eta}} = \hat{\alpha} + \mathbf{X}'\hat{\boldsymbol{\beta}}$.

D.1 Descent convergence

Despite the non-differentiability of $|\beta_j|$ at zero, Tseng (2001) establishes local convergence for CD on (34) as a consequence of penalty separability: the non-differentiable part of our objective is a sum of functions on only a single coordinate. Thus CD solves each weighted-least squares problem, and the full algorithm converges if IRLS does. For non-Gaussian models, convergence of such nested L_1-penalized IRLS algorithms is shown in Lee et al. (2014).

D.2 Quasi-Newton acceleration

Under high collinearity and large γ, one may wish to accelerate convergence via a quasi-Newton step (e.g., Lange, 2010). Acceleration is applied to $\boldsymbol{\theta} = [\alpha, \boldsymbol{\beta}]$, and a move is accepted only if it leads to a decrease in the objective. Suppose that $\hat{\boldsymbol{\theta}}^{(0)}$, $\hat{\boldsymbol{\theta}}^{(-1)}$, and $\hat{\boldsymbol{\theta}}^{(-2)}$ are the current, previous, and previous-to-previous parameter estimates. Write $M(\hat{\boldsymbol{\theta}}^{(t)})$ as the implied CD update map $\hat{\boldsymbol{\theta}}^{(t)} \to \hat{\boldsymbol{\theta}}^{(t+1)}$, such that the algorithm converges at $\hat{\boldsymbol{\theta}} - M(\hat{\boldsymbol{\theta}}) = \mathbf{0}$. With $\mathbf{u} = \hat{\boldsymbol{\theta}}^{(-1)} - \hat{\boldsymbol{\theta}}^{(-2)}$ and $\mathbf{v} = \hat{\boldsymbol{\theta}}^{(0)} - \hat{\boldsymbol{\theta}}^{(-1)}$, a secant approximation to the gradient of M is $\partial M/\partial\hat{\theta}_l \approx \text{v}_l/\text{u}_l$. An approximate Newton-Raphson step to solve for the root of $\hat{\boldsymbol{\theta}} - M(\hat{\boldsymbol{\theta}})$ updates each coordinate $\hat{\theta}_l \leftarrow \hat{\theta}_l^{(-1)} - (\hat{\theta}_l^{(-1)} - \hat{\theta}_l^{(0)})/(1 - \text{v}_l/\text{u}_l)$ which can be re-written as $\hat{\theta}_l = (1 - \text{w}_l)\hat{\theta}_l^{(-1)} + \text{w}_l\hat{\theta}_l^{(0)}$ where $\text{w}_l = \text{u}_l/(\text{u}_l - \text{v}_l)$.