One-step estimator paths for concave regularization

Matt Taddy

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. The methods 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 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' (either asymptotically or with high probability). From an information compression perspective, the increased sparsity encouraged by diminishing-bias penalties (since single large coefficients are allowed to account for the signals of other correlated covariates) leads to lower memory, storage, and communication requirements. Such savings are very important in distributed computing schemes (e.g., Taddy, 2015).

Unfortunately, exact solvers for concave 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 combines ideas from OSE with the concept of a *regularization path* – a general technique, most

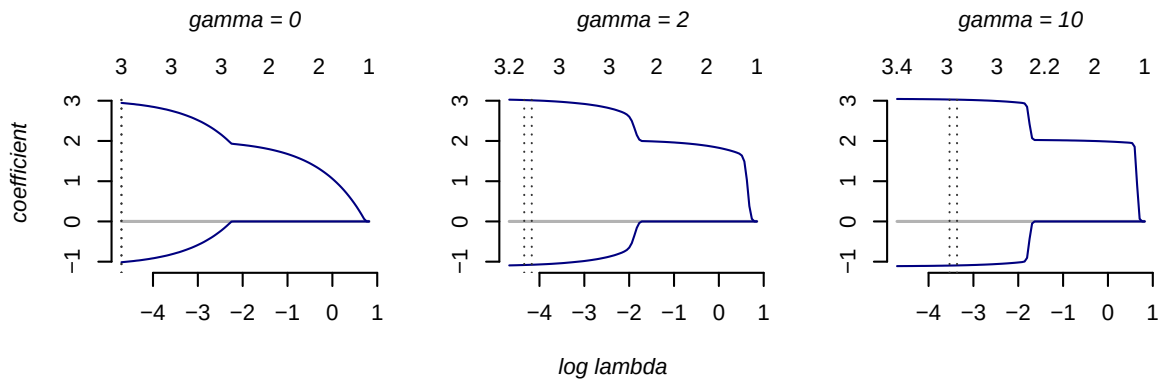

Figure 1: Gamma lasso estimation on $n = 10^3$ observations of $y_i = 4 + 3x_{1i} - x_{2i} + \varepsilon_i$, where $\varepsilon_i \overset{ind}{\sim} \mathrm{N}(0, 1)$ and $\{x_{1i}, x_{2i}, x_{3i}\}$ are marginally standard normal with correlation of 0.9 between covariates (x_{3i} is spurious). The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

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 the 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 novel 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 reliable 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* im-

plementation of POSE from which we are able to construct heuristic 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 $\boldsymbol{x}_j = [x_{1j}, \ldots, x_{nj}]'$.[1] Write $\eta_i = \alpha + \mathbf{x}_i'\boldsymbol{\beta}$ as the linear equation for observation i, and denote with $l(\alpha, \boldsymbol{\beta}) = l(\boldsymbol{\eta})$ 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 the solution to

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

where $\lambda > 0$ controls overall penalty magnitude and $c()$ 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-

[1] 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.

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.

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 proceed down to some pre-specified λ^T (e.g., $\lambda^T = 0.01\lambda^1$).

Our path of one-step estimators (POSE) framework is in Algorithm 1. 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 $b \neq 0$.

Algorithm 1 POSE

Initialize $\lambda^1 = \inf\left\{\lambda : \ \hat{\boldsymbol{\beta}}|_\lambda = \mathbf{0}\right\}$, so that $\hat{\boldsymbol{\beta}}_1 = \mathbf{0}$.

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

for $t = 2 \ldots T$:

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

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

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

Section 3 will detail how POSE relates to concave regularization. However, for some quick intuition, consider POSE with a concave cost function (such as the log penalty in Figure 2). The derivative $c'(|\hat{\beta}|)$ 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,

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

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. 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 (2013) and Armagan et al. (2013).

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 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 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.

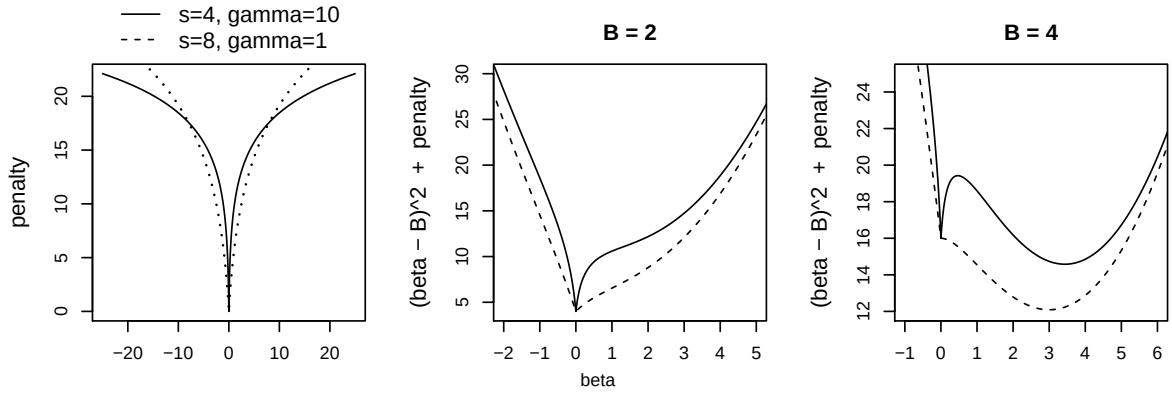

Figure 2: Log penalties $c(\beta) = s\log(1+\gamma|\beta|)$ and penalized objectives $(\beta - B)^2 + c(\beta)$.

Unfortunately, such penalties can overwhelm the convex likelihood and produce a concave 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 solution over a large set of specifications[2] 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 concave 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), a technique inspired by Bickel (1975) that amounts to taking 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

[2]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.

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 regression coefficients $\hat{\beta}_j^0 = \mathrm{cor}(\mathbf{x}_j, \mathbf{y})$.

The main point here 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. 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 penalized estimation

Our oracle benchmark is estimation under L_0 costs, $c(\beta_j) = \mathbb{1}_{\{\beta_j \neq 0\}}$, for which global solution is impractical. We are interested in weighted-L_1 penalization as a way to obtain fits that are as sparse as possible without compromising predictive ability, regardless of the underlying data generating process (or 'true' sparsity).

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 $\beta_j^S = 0 \; \forall \; j \notin S$. 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, with proof in the supplemental appendix.

LEMMA 3.1. *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* $j \in 1 \dots p$,

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

In addition, we need to define *restricted eigenvalues* (RE) on the gram matrix $\mathbf{X}'\mathbf{X}/n$.[3]

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 rules from L_0 and weighted-L_1 penalized estimation.

THEOREM 3.1. *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*

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

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

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

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

Since $\hat{\mathbf{y}}'\mathbf{e}^S = \hat{\mathbf{y}}'(\mathbf{I} - \mathbf{H}^S)\mathbf{y} = \hat{\boldsymbol{\beta}}'\mathbf{X}'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\nu) = \sum_{j\in S^c}\hat{\beta}_j\boldsymbol{x}_j'(\mathbf{y} - \mathbf{X}\boldsymbol{\beta}^\nu)$, we can apply Lemma 3.1 followed by $\boldsymbol{\beta}^\nu$ being optimal under L_0 penalty ν to get

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

[3]This RE matches the 'adaptive restricted eigenvalues' of Buhlmann and van de Geer (2011). Similar quantities are common in the theory of regularized estimators; see also Raskutti et al. (2010) and Bickel et al. (2009).

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 (7),

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

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

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

Dividing each side by $\sqrt{n}\|\mathbf{X}(\hat{\boldsymbol{\beta}} - \boldsymbol{\beta}^\nu)\|/2$ 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 $L = \frac{\|\boldsymbol{\omega}_S\|}{\sqrt{s}}\left(\omega^{\min}_{S^c} - \sqrt{2}\sigma/(\lambda\sqrt{n})\right)^{-1}$, 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 $\omega^{\min}_{S^c} > (\sigma/\lambda)\sqrt{2/n}$, 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 $\omega^{\min}_{S_c}$ 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 over-fit. For this reason it is essential that we have tools for choosing optimal λ.
- In the supplemental material, we also 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. However, such results depend upon design restrictions that are less realistic in practice.

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: they are too large to store on a single machine. 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. Such instability arises for large γ combined with small λ in our GL algorithm and, more generally, in many concave regularized estimators; see the supplemental material for a detailed 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 then combined with the fitted deviance for penalty selection via common information criteria.

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

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

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

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

and is straightforward to show (supplemental) that this is equivalent to the log-penalized objective

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

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 $\hat{\beta}_j = 0$.[4]

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 $df = \hat{p}^t$ 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 $k(df) = df \times n/(n - df - 1)$

[4]Note that the number of unpenalized variables (e.g., 1 for α) should also be added to the total estimation df.

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 samples of size $n = 1000$ from

$$y \sim \mathrm{N}\left(\mathbf{x}'\boldsymbol{\beta}, \sigma^2\right) \text{ where } \beta_j = \frac{\exp\left(-\frac{j}{\mathsf{d}}\right)}{j},\ j = 1 \dots p,$$
$$\text{and } \mathbf{x} = \mathbf{u} * \mathbf{z},\ \ \mathbf{u} \sim \mathrm{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right),\ \ z_j \overset{ind}{\sim} \mathrm{Bin}(0.5). \tag{17}$$

Each simulation draws means $\eta_i = \mathbf{x}_i'\boldsymbol{\beta}$, and two response samples (train and test) $\mathbf{y}, \tilde{\mathbf{y}} \sim \mathrm{N}(\boldsymbol{\eta}, \sigma^2\mathbf{I})$. Multicollinearity is parametrized via $\Sigma_{jk} = \rho^{|j-k|}$ with $\rho = 0.5$. We define σ^2 through *signal-to-noise* ratios $\mathrm{sd}(\boldsymbol{\eta})/\sigma$ of $1/2$, 1, and 2; for coefficient decay rates we consider d of 10, 50, and 100.

The system in (17) 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

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

which is solvable here by searching through OLS on $\mathbf{X}_{\{1\dots j\}}$ for $j = 1 \dots p$ (since coefficients are ordered).

Table 1 shows predictive performance, measured through out-of-sample (OOS) $R^2 = 1 - \mathrm{var}(\tilde{\mathbf{y}} - \hat{\mathbf{y}})/\mathrm{var}(\tilde{\mathbf{y}})$, for `gamlr` runs of GL with γ of 0 (lasso), 2, and 10. For penalty selection, we use AICc with the GL df from Section 4. Results are compared to the oracle as well as `sparsenet`'s MCP penalized regression, which applies 5-fold CV to optimize out-of-sample error over a large grid of potential penalty sizes (λ) and concavities (analogous to our γ).

Note that Table 1 represents only a small portion of our simulation study. The supplemental material includes results for hundreds of additional configurations: varying levels of sparsity (different z_i distribution) or multicollinearity (different $\boldsymbol{\Sigma}$) in $\mathbf{x}$, as well as for different prediction algorithms: various CV selection rules, AIC, and BIC applied to the GL paths as well as

$\mathrm{sd}(\boldsymbol{\eta})/\sigma$	d	ρ	% Worse than oracle C_p									C_p R^2
			lasso		GL $\gamma = 1$		GL $\gamma = 10$		marginal AL			
			AICc	*CV*	*AICc*	*CV*	*AICc*	*CV*	*AICc*	*CV*	MCP	
		0	3	3	2	2	4	3	2	2	**1**	*0.79*
	10	*0.5*	5	5	4	3	5	5	8	8	**2**	*0.79*
		0.9	7	6	5	5	7	6	10	10	**3**	*0.79*
		0	6	5	5	**4**	6	7	5	5	**4**	*0.77*
2	*50*	*0.5*	11	8	8	7	10	10	14	13	**5**	*0.77*
		0.9	14	10	10	8	12	11	45	45	**7**	*0.77*
		0	9	6	7	6	7	8	8	7	**5**	*0.75*
	100	*0.5*	18	9	11	**8**	12	11	21	17	9	*0.75*
		0.9	23	11	13	**9**	14	12	56	56	10	*0.75*
		0	8	8	7	6	14	9	9	10	**5**	*0.48*
	10	*0.5*	16	16	12	11	20	16	17	17	**6**	*0.48*
		0.9	22	22	18	17	27	21	**9**	**9**	10	*0.48*
		0	19	**17**	**17**	18	25	26	**17**	18	**17**	*0.44*
1	*50*	*0.5*	38	32	28	28	30	40	39	38	**25**	*0.44*
		0.9	69	68	57	46	70	59	46	46	**31**	*0.44*
		0	26	**20**	24	30	33	51	22	22	**20**	*0.40*
	100	*0.5*	64	57	**42**	66	45	95	51	47	56	*0.40*
		0.9	86	88	89	87	93	89	**65**	66	85	*0.40*
		0	26	27	28	**23**	59	30	43	57	**23**	*0.18*
	10	*0.5*	49	51	44	41	79	55	69	76	**28**	*0.18*
		0.9	44	44	43	42	49	43	**28**	**28**	39	*0.18*
		0	**53**	56	89	82	101	91	72	91	57	*0.13*
0.5	*50*	*0.5*	**90**	94	99	98	101	99	106	117	94	*0.13*
		0.9	78	79	80	78	88	79	82	80	**75**	*0.13*
		0	**61**	67	106	98	101	100	85	115	68	*0.09*
	100	*0.5*	**94**	97	101	101	101	101	118	132	98	*0.09*
		0.9	**86**	88	91	88	98	90	95	92	87	*0.09*

Table 1: Average out-of-sample R^2, reported as % worse than the C_p oracle, across 1000 different samples from (17). GLγ columns denote `gamlr` with AICc penalty selection, while MCP is `sparsenet` with 5-fold CV selection. The best results are bolded.

to additional adaptive lasso schemes. Moreover, the supplement includes results on sensitivity and false discovery rates with respect to the support used in the oracle C_p scheme, along with plots for coefficient estimates, information criteria, and OOS error rates along λ.

Insights from the simulations (including supplement):

- One of the three GL γ-specifications is always able to provide performance near to (never more than 1.5% worse, and sometimes better) that of the computationally intensive routine of CV-selected MCP (over a grid in penalty size and scale). A single `gamlr` run requires a fraction of a second; `sparsenet` with 5-fold CV requires 15-20 seconds per run.

- AICc and 5-fold CV give roughly the same performance for the GL0 (lasso) and GL2; AICc is often much better than CV for the less stable GL10 paths. For prediction, AICc is a massive improvement over either BIC or the uncorrected AIC; the latter leads to negative R^2 in many configurations.

- Selected GL2 fits had often half the number of nonzero coefficients as for the selected lasso fit; GL10 leads to even more sparsity, sometimes returning less than 10% of the number of nonzero coefficients given by lasso. The results suggest that running `gamlr` with small γ (e.g., as in GL2) can provide much additional information compression without hurting performance. Even the very sparse $\gamma = 10$ fits do nearly as well as best except in the low signal settings, where all but lasso suffer. Increasing γ also leads to lower false discovery rates with respect to the C_p oracle.

- As γ increases, the CV error increases more quickly after it's minimum; this is predicted by Theorem 3.1, and indicates that the consequences of over-fit are worse under faster diminishing-bias. The computation times also increase, although a single run at $\gamma = 10$ still requires less than a second.

- Investigation of AICc values finds that minimization of this criteria *across* GL0 and GL2 would have consistently chosen the rule that does best in Table 1. Thus AICc selection on just two GL paths, with combined compute time still less than a second, yields the best or near best predictor in all scenarios.

	$\gamma = 0$			$\gamma = 1$			$\gamma = 10$		
		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
		305 nonzero effects			*204 nonzero effects*			*64 nonzero effects*	

Table 2: Top 10 AICc selected player 'partial plus-minus' (PPM) values for the 2013-2014 season.

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. The AICc surface is shown in Figure 3; it selects denser models with low λ but not-to-big

γ. See the supplement for additional study, including comparison to BIC and CV.

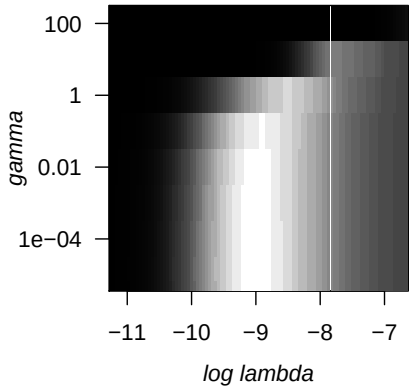

Figure 3: Hockey AICc, rising from white to black.

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 $p_j = e^{\beta_j}/(1+e^{\beta_j})$. 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.

Table 2 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 (see supplement), 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

Concave penalized estimation for Big Data, where exact solvers are too expensive, 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 diminishing-bias sparse compressed sensing no more cost than a standard lasso. We know of no other software that meets this standard.

References

Akaike, H. (1973). Information theory and the maximum likelihood principle. In *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.

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). 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.

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.

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*.

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*.

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*(5), 2183–2202.

Wang, L., Y. Kim, and R. Li (2013). 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*(5B), 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*(4), 1509–1533.