The Gamma Lasso

Matt Taddy

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

This article describes a very fast algorithm for obtaining continuous regularization paths corresponding to cost functions spanning the range of concavity between L_0 and L_1 norms. The 'gamma lasso' heuristic does L_1 (lasso) penalized regression estimation on a grid of decreasing penalties, but adapts coefficient-specific weights to decrease as a function of the estimated coefficient in the previous path segment. Our particular weight-updating scheme is motivated from a Bayesian model, and is related to estimation under log penalties. This very simple recipe is used to illustrate the large and difficult literature on concave penalization, with the hope that we can make the ideas more accessible to practitioners. The construction also leads us to a plug-in estimator for degrees of freedom; this is applied in model selection, and in experimentation our information criteria perform as well as cross-validation. The work is illustrated in linear regression simulations and in application of logistic regression to evaluate hockey players.

1 Introduction

For regression in high-dimensions it is useful to regularize estimation with a penalty on coefficient size. One minimizes the negative log likelihood plus a sum of cost functions applied to each coefficient's distance from zero. When these cost functions have a non-differentiable spike at zero, the optimal solution can be sparse. The curvature of the penalty away from zero dictates the weight of shrinkage imposed on the nonzero coefficients: zero curvature is an L_1 penalty, as used in the common lasso regression framework (Tibshirani, 1996), and as curvature goes towards $-\infty$ one approaches the L_0 penalty of subset selection.

Imagine that you are a data analyst familiar with classical methods, such as hypothesis testing, looking to try some of these more modern sparse regularized regression techniques. Strictly concave penalties, with negative curvature instead of lasso's zero second derivative, are tempting because they provide near-unbiased estimation for large signals. That is, if $\hat{\beta}_j \neq 0$ then it will be close to the result for maximum likelihood estimation (MLE) on *in-the-model* coefficients. Unfortunately, you will discover that computation is difficult with these strictly concave penalties. Most available software takes far longer to run than convex alternatives and the solvers are not generally globally convergent. At the extreme, solving for L_0 regularized coefficients is NP hard. In contrast, the lasso has a very fast forward stepwise procedure (Efron et al., 2004), easy-to-estimate degrees of freedom (Zou et al., 2007), and is far more widely used in practice. These conveniences are due to the estimator stability that results from lasso's nondiminishing bias. How do you decide what to use, and is it possible to quickly evaluate the range of options between L_0 and L_1 costs?

We introduce in Section 2 the 'gamma lasso' (GL) algorithm. The purpose of the article is then twofold. First, to advertise a very fast and stable estimation framework that inherits many of the desirable properties of concave penalized estimation: algorithm details are in Section 6, simple Monte-Carlo and information criterion model selection are in Section 9, and all methods are implemented in the `gamlr` package for R. Second, by reading the large and difficult literature on concave regularization in the context of our simple algorithm, we seek to provide an intuitive overview on practical application of these techniques. This includes Bayesian interpretations in Section 3, consistency and unbiasedness properties in Section 4, types and benefits of estimator stability in Section 5, available estimation routines in Section 7,

and model complexity (degrees of freedom) in Section 8.

This mixture of goals follows from our personal use of the GL routine, both as a workhorse in applications and as a tool for explaining non-convex penalized estimation to non-experts. The ideas are illustrated via linear regression simulations in Section 10 and in logistic regression to evaluate the ability of hockey players in Section 11. Section 12 closes with a brief discussion.

2 The gamma lasso

Denote the data matrix of p covariates for n observations as $\mathbf{X} = [\mathbf{x}_1 \cdots \mathbf{x}_n]'$, where $\mathbf{x}_i = [x_{i1}, \ldots, x_{ip}]'$, and the associated response as $\mathbf{y} = [y_1, \ldots, y_n]'$. Since the size of penalized β_j depends upon the units of x_{ij}, it is common to scale the coefficient by sd_j, the standard deviation of the j^{th} column of $\mathbf{X}$; this is achieved if x_{ij} is replaced by x_{ij}/sd_j throughout. Write $\eta_i = \alpha + \mathbf{x}_i'\boldsymbol{\beta}$ as the linear equation for observation i, and denote with $l(\alpha, \boldsymbol{\beta}) = l(\boldsymbol{\eta})$ an unregularized objective proportional to the negative log likelihood. In Gaussian (linear) regression with independent normal errors, $l(\boldsymbol{\eta})$ 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 $y_i \in [0, 1]$.

The gamma lasso in Algorithm 1, for 'scale' $\gamma \geq 0$, yields paths of penalized coefficient estimates $\hat{\boldsymbol{\beta}}^1 \ldots \hat{\boldsymbol{\beta}}^T$ (and intercepts $\hat{\alpha}^1 \ldots \hat{\alpha}^T$) corresponding to L_1 penalties $\lambda^1 > \lambda^2 \ldots > \lambda^T$ multiplied against adaptive coefficient-specific weight adjustments $\boldsymbol{\omega}^1 \ldots \boldsymbol{\omega}^T$.

Algorithm 1 The gamma lasso

Initialize $\boldsymbol{\omega}^1 = \mathbf{1}$ and $\lambda^1 > 0$ with step size $0 < \delta < 1$.

for $t = 1 \ldots T$:

$$\left[\hat{\alpha}, \hat{\boldsymbol{\beta}}\right]^t = \underset{\alpha, \beta_j \in \mathbb{R}}{\operatorname{argmin}} \; l(\alpha, \boldsymbol{\beta}) + n \sum_j \lambda^t \omega_j^t |\beta_j| \tag{1}$$

$$\omega_j^{t+1} = \left(1 + \gamma |\hat{\beta}_j^t|\right)^{-1} \quad j = 1 \ldots p \tag{2}$$

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

Outputs from this type of algorithm – a $p \times T$ field of $\hat{\boldsymbol{\beta}}$ estimates obtained while moving from high to low penalization – are referred to as *regularization paths*. LARS (Efron et al., 2004) is a well-known example. The approach is nice not only because it leads to a full range

of candidates for model selection (as in Section 9), but also because $\hat{\boldsymbol{\beta}}^t$ at λ^t provide *hot-starts* for $\hat{\boldsymbol{\beta}}^{t+1}$ at the next penalty level. If the paths are near continuous (see Section 5), estimation over the full path can be faster than cold-start solution for a single penalty specification.

To start such paths, λ^1 is commonly set to infimum λ such that (1) is minimized at $\hat{\boldsymbol{\beta}} = \mathbf{0}$. Write $g_j(\hat{\boldsymbol{\beta}})$ for the j^{th} coefficient gradient of $l(\alpha, \boldsymbol{\beta})$ (see A.1) evaluated at estimates $[\hat{\alpha}, \hat{\boldsymbol{\beta}}]$, where it is implied that $\hat{\alpha}$ has been set to minimize $l(\alpha, \hat{\boldsymbol{\beta}})$. The initial value is available analytically as

$$\lambda^1 = n^{-1} \max\{|g_j(\mathbf{0})|,\ j = 1 \dots p\}, \tag{3}$$

the maximum mean absolute gradient for the null model with $\boldsymbol{\beta} = \mathbf{0}$. Given a path-length T, λ^T is specified as a ratio of λ^1 (e.g., $\lambda^T = 0.01\lambda^1$) and the step-size δ is implied as a consequence.

Behavior of the GL paths along a given $\boldsymbol{\lambda}$ sequence is governed by γ, which we refer to as the penalty scale (see Section 3). Under $\gamma = 0$ the weights are $\omega_j^t = 1$ for all j, t and Algorithm 1 is just the usual lasso. At the other 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 as a function of λ^t in a simple problem. Moving from the lasso ($\gamma = 0$) to more concavity (larger γ), the estimates are less shrunk towards zero. As a result, selection of models along the $\boldsymbol{\lambda}$ paths becomes less stable: small specification jitter implies larger changes in $\hat{\boldsymbol{\beta}}$ for a chosen λ (however note that even at $\gamma = 10$ the paths appear to be continuous).

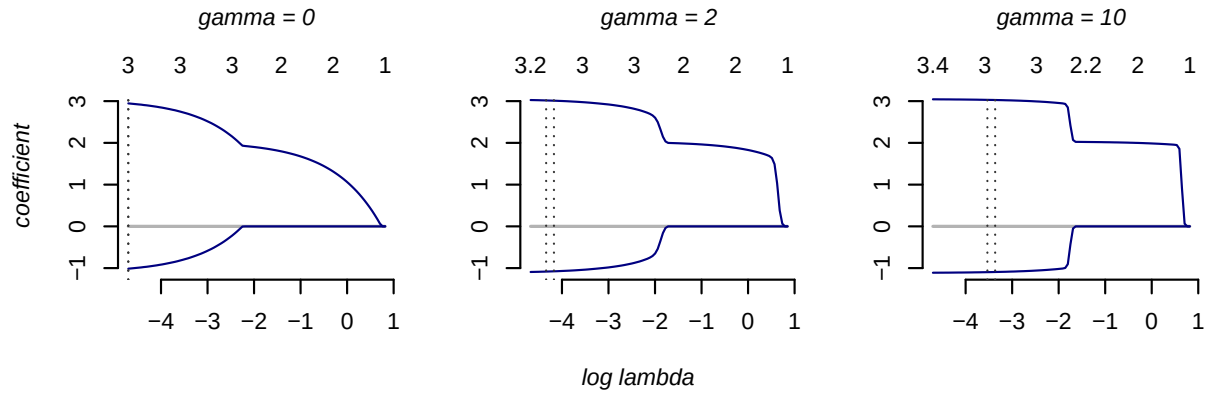

Figure 1: Gamma lasso estimation on $n = 10^3$ observations of $y_i = 4 + 3x_{1i} - x_{2i} + \varepsilon_i$, where $\varepsilon_i \overset{ind}{\sim} \mathrm{N}(0, 1)$ and $\{x_{1i}, x_{2i}, x_{3i}\}$ are marginally standard normal with correlation of 0.9 between covariates (x_{3i} is spurious). The penalty path has $T = 100$ segments, $\lambda^1 = n^{-1} |\sum_i x_{1i} y_i|$, and $\delta = 0.01^{1/99}$. Degrees of freedom are on top and vertical lines mark AIC and BIC selected models (see Sections 8-9).

3 Bayesian motivation

Consider the Bayesian lasso (Park and Casella, 2008), where each β_j is assigned an independent Laplace distribution prior with scale $\tau_j > 0$,

$$\beta_j \sim \mathrm{La}\,(\tau_j) = \frac{\tau_j}{2} \exp\left[-\tau_j |\beta_j|\,\right]. \tag{4}$$

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 (4) is then lasso estimation. In a fully Bayesian analysis, λ is assigned a hyperprior and coefficient estimates are integrated over its posterior.

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

$$\pi(\beta_j, \tau_j) = \mathrm{La}\,(\beta_j;\ \tau_j)\,\mathrm{Ga}\left(\tau_j;\ s, \gamma^{-1}\right) = \frac{1}{2\Gamma(s)} \left(\frac{\tau_j}{\gamma}\right)^s \exp\left[-\tau_j(\gamma^{-1} + |\beta_j|)\right]. \tag{5}$$

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

Write $s = n\lambda/(\gamma\phi)$, such that $\mathbb{E}[\tau_j] = n\lambda/\phi$ and $\mathrm{var}(\tau_j) = \gamma\mathbb{E}[\tau_j]$. Then the MAP scale estimate is $\hat{\tau}_j = \omega_j(n\lambda/\phi)$ with $\omega_j = (1 + \gamma|\beta_j|)^{-1}$, and the gamma lasso of Algorithm 1 appears through a sequence of MAP estimates under the joint prior in (5).

At each λ^t:

- use the most recent coefficient estimate to fix $\hat{\tau}_j^t = (n\lambda^t/\phi)/(1 + \gamma|\beta_j^{t-1}|)$.
- find $\hat{\boldsymbol{\beta}}^t$ to maximize the posterior under $\mathrm{La}(\hat{\tau}_j^t)$ coefficient priors.

While clearly easier to compute, this sequential 'greedy' MAP seems a poor cousin to an actual joint MAP estimate – i.e., that which maximizes the posterior for both $\boldsymbol{\tau}$ and $\boldsymbol{\beta}$. For example, our estimates are sensitive to path step-size: as $\delta \to 1$ GL approaches the joint MAP, and it gets further from this solution as δ decreases. However, we'll see later that hedging away from joint optimality has useful stabilization effects in addition to making for much faster analysis. First, we need to look a bit deeper at MAP estimation under our Bayesian model.

3.1 Log penalization

Consider joint MAP estimation of $[\boldsymbol{\tau}, \boldsymbol{\beta}]$ under the prior in (5), where we've suppressed α for simplicity. By taking negative logs and removing constants, this is equivalent to solving

$$\min_{\beta_j \in \mathbb{R},\ \tau_j \in \mathbb{R}^+} \quad \phi^{-1} l(\boldsymbol{\beta}) + \sum_j \left[\tau_j (\gamma^{-1} + |\beta_j|) - s \log(\tau_j) \right]. \tag{6}$$

By concentrating-out of $\boldsymbol{\tau}$, it is straightforward to show that (6) is equivalent to the objective

$$\min_{\beta_j \in \mathbb{R},\ \tau_j \in \mathbb{R}^+} \quad \phi^{-1} l(\boldsymbol{\beta}) + \sum_j s \log(1 + \gamma |\beta_j|) \tag{7}$$

PROPOSITION 3.1. *$\hat{\boldsymbol{\beta}}$ solves (7) if and only if it is also in the solution to (6).*

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 (6) 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 (6) and removing constant terms yields (7). Moreover, the solution to (6) solves (7): otherwise, there would need to be a point on the profile slice of (6) defined by $\tau_j = \tau(\hat{\beta}_j)$ that is lower than its minimum.

Cost function $c(\beta_j) = s \log(1 + \gamma |\beta_j|)$, where $s, \gamma > 0$, is referred to as the log penalty; it is concave with curvature $-s/(\gamma^{-1} + |\beta_j|)^2$ and 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 (2013b) and Armagan et al. (2013). The penalty is illustrated in Figure 2.

3.2 Generalized double Pareto priors

For a Bayesian it is odd to be solving for $\boldsymbol{\tau}$ rather than marginalizing over its uncertainty. However, recognizing the functional form of a gamma density in (5), $\pi(\beta_j, \tau_j)$ integrates over τ_j to yield the marginal prior $\pi(\beta_j) = 0.5 s \left(1 + \gamma |\beta_j|\right)^{-(s+1)}$. 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 (6), which is also the log penalized estimator from (7), gains additional interpretation as the *marginal* MAP for $\boldsymbol{\beta}$ under $\mathrm{Ga}(s - 1, 1/\gamma)$ hyperpriors on each τ_j.

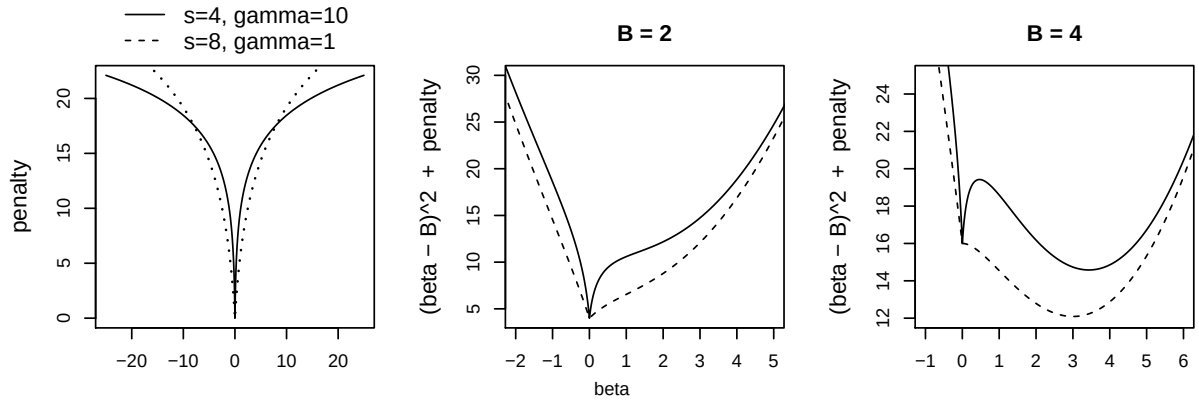

Figure 2: Log penalties $c(\beta) = s\log(1+\gamma|\beta|)$ and penalized objectives $(\beta - B)^2 + c(\beta)$.

4 Diminishing bias

The defining distinction between the gamma lasso and the standard lasso is that cost gradient for each β_j is decreasing as a function of the size of its effect on the likelihood. This occurs algorithmically: if the signal is strong enough that $\hat{\beta}_j^t$ is nonzero at λ_t, the penalty at segment $t+1$ is deflated by the factor $1/(1+\gamma|\beta_j^t|)$ at the next path segment (see also (8), the GL two-step thresholding operator). Consider the log penalty that occurs as $\lambda^t \to \lambda^{t-1}$. Under our GL parametrization, the log penalty curvature is $-n\lambda\gamma/(1+\gamma|\beta_j|)^2$: it is strictly decreasing with $|\beta_j|$ and goes to zero as $|\beta_j| \to \infty$. This can be observed in the left panel of Figure 2, and larger γ causes the penalty to flatten faster.

We refer to this property of diminishing penalty curvature as 'unbiasedness for large signals'. It is *the* reason why one would use concave penalization instead of L_1 or convex alternatives. Theoretically, it is the primary necessary condition for a penalized estimator to have oracle properties: a class of results showing that coefficient estimates under large n will be the same as if you knew which should be zero. Fan and Li (2001) introduce the framework, Fan and Peng (2004) allow p to grow slowly with n, and Armagan et al. (2013) provide results specific to log penalization. Oracle properties make strong assumptions about near-sparsity of true $\boldsymbol{\beta}$, but they form a popular approximation framework for evaluating high-dimensional estimators.

There are also a variety of practical reasons for seeking diminishing bias. First, having large signals estimated without attenuation can increase model sparsity. This is useful from a purely computational perspective whenever storage is an issue, say, with distributed massive data analysis (Taddy, 2013a). In causal inference procedures, where penalized projection from

$\mathbf{x}$ on y is intended to control for confounding variables rather than for pure prediction, Belloni et al. (2012) advocate use of unbiased coefficient estimates. Finally, many practitioners familiar with classical methods (and versions of subset selection) will seek out low-bias estimates, regardless of the application.

5 Stability and Continuity

The benefits of Section 4 come at a price: if the penalty is concave, the negative log posterior is not necessarily convex. This can be seen in the right two panels of Figure 2: the objectives are concave approaching the origin, and can even become bi-modal.

The obvious effect of a non-convexity is computational: the models take longer to fit. The genius of LARS and the lasso is that entire regularization paths are easy to calculate *because* estimates change slowly along those paths. In contrast, the previous segment $\hat{\boldsymbol{\beta}}^{t-1}$ will not be a good hot-start for $\hat{\boldsymbol{\beta}}^{t}$ if the solution paths have discontinuity jumps. This causes a corresponding jump in computation time; for example, Figure 3 shows timings growing rapidly after this threshold (around $1 < \gamma < 10$) for the hockey data of Section 11. Here the cost difference is mere seconds, but in larger n and p applications the jumps become prohibitively expensive.

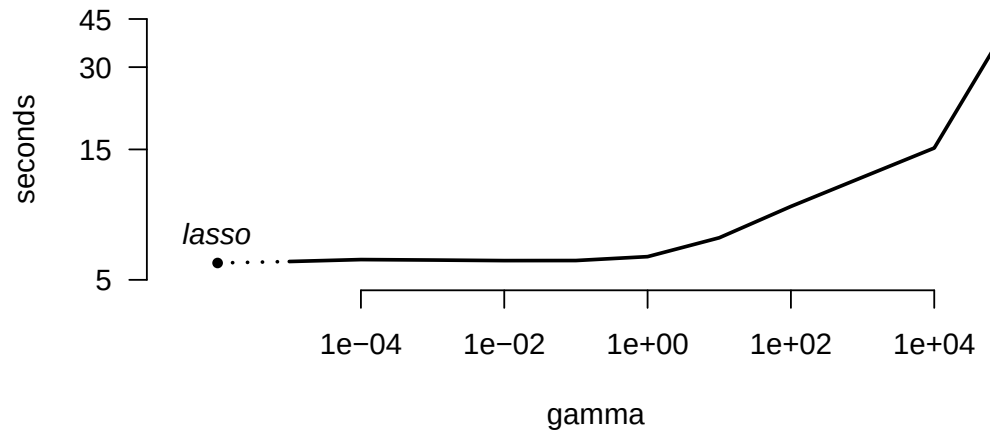

Figure 3: Timings for the hockey data fits of Section 11 on a length-100 grid with $\lambda^{100} = 0.01\lambda^{1}$. This used a high absolute convergence tolerance of 10^{-10}; results are less dramatic under default settings.

Such discontinuities occur if the objective is concave at the origin. Consider the solid line in the right panel of Figure 2: under small permutation to penalty γs, the solution moves between 0 and near 4. When conditional updates in coordinate descent (see Section 6) have this

shape, all other dimensions of the model fit need to change dramatically depending upon such thresholds. Moreover, finding a *global* solution is computationally intractable if the objective is non-convex; this issue is discussed in detail in Section 7.

A more subtle possible consequence of concavity is estimator instability. Path discontinuities will cause jumps in the implicit estimation function for $\hat{\boldsymbol{\beta}}$, such that small data jitter leads to large changes in model fit. As every statistics student should know, mean squared error for $\hat{y}$ as a predictor of y can be written as the sum of squared bias $(y - \mathbb{E}[\hat{y}])^2$ plus estimator variance $\mathrm{var}(\hat{y})$. Thus instability leads to lower quality prediction due to a large increase in the sampling variance. As detailed in the classic Breiman (1996) discussion, the effects of instability are especially dramatic when doing model selection, since variability of this choice explodes with the estimation variance for each individual candidate model (see also Section 9).

Given these issues, the benefits of concave penalties (i.e., the diminishing bias of Section 4) are only worthwhile so long as we are able to remain stable. The rest of this section looks first at what this means in terms of our Bayesian model and second at how sub-optimality of the gamma lasso as a joint MAP actually stabilizes the procedure.

5.1 Prior variance

For orthogonal inputs, the penalized objective becomes concave only if negative log likelihood coordinate curvature $h_j = \partial^2 l(\boldsymbol{\beta})/\partial \beta_j^2$, which does not depend upon y for exponential families, is less than the absolute value of the curvature on the cost function for β_j. Absolute curvature for the log penalty is $s\gamma^2$ at the origin, and less elsewhere, such that the objective in (7) is guaranteed convex if $s\gamma^2 < h_j$. For correlated inputs, one has objective convexity if the minimum eigenvalue of $\mathbf{H}$, the Hessian matrix of second derivatives of $l(\boldsymbol{\beta})$, is greater than $s\gamma^2$.[1]

Recognizing $\gamma^2 s$ as the variance of a $\mathrm{Ga}(s, \gamma^{-1})$ distribution, the above can be given a clear Bayesian interpretation in the context of our model from Section 3: the joint MAP is stable (changes little under small data jitter) if the prior variance on each τ_j coefficient scale is less than the minimum eigenvalue of the likelihood information matrix $\mathbf{H}$. For Gaussian regression with standardized orthogonal covariates, $h_j = \sum_i x_{ij}^2 = n$ and we get a simple rule: the joint

[1] 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 $\mathbf{H} - s\gamma^2\mathbf{I}$ and $(\mathbf{H} - s\gamma^2\mathbf{I} + s\gamma^2\mathbf{I} - \nu\mathbf{I})\mathbf{v} = 0$ so that $\nu - s\gamma^2$ is an eigenvalue of the minimization objective.

MAP is stable if prior variance on each τ_j is less than the number of observations. For logistic regression you need $\mathrm{var}(\tau_j) < n/4$ to get the same guarantee. And for marginal rather than joint MAP estimation you need $\mathrm{var}(\tau_j) + \gamma^2$ less than n or $n/4$.

5.2 Stability of the gamma lasso

A rigorous definition of stability can be expressed in terms of the continuity of estimators as a function of $\mathbf{y}$, the observed response. In particular, Lipschitz continuity of such functions is of primary importance in estimation; e.g., for unbiased risk assessment as in Stein (1981). Many popular concave cost functions, such as the SCAD penalty of Fan and Li (2001) and the MC+ penalty of Zhang (2010), have been engineered to be Lipschitz continuous (at least under certain parameterizations). It turns out that gamma lasso estimators have this property.

Say $f(b)$ is the implicit function for $\hat{\beta}$ as a function of the MLE b. f is said to be Lipschitz continuous if $|f(b_1) - f(b_2)| \leq L|b_1 - b_2|$ for some finite constant L on all b_1, b_2 in the domain of f. If the MLE b is itself Lipschitz with respect to $\mathbf{y}$, then we'll say that the estimator is Lipschitz stable. Consider the single-coefficient two step path $[\hat{\beta}^0, \hat{\beta}^1]$ on $[\lambda, \delta\lambda]$ for $0 < \delta < 1$, such that $\hat{\beta}^0$ is the lasso estimate under $\tau = \lambda$ and $\hat{\beta}^1$ is the lasso estimate under $\tau = \delta\lambda/(1 + \gamma|\hat{\beta}^0|)$. Given MLE b we can re-write $\hat{\beta}^1$ as

$$f(b) = \left(b - \frac{\delta\lambda}{1 + \gamma(b - \lambda)_+} \right)_+ \tag{8}$$

where $\gamma, \lambda, \delta > 0$, $\lambda < \infty$, $\delta < 1$. We call this the gamma-lasso thresholding operator.

PROPOSITION 5.1. *For $\gamma < \infty$ the GL thresholding operator in (8) is Lipschitz continuous.*

Proof. Take two arbitrary MLE $b_1 > b_2 > 0$; other configurations work similarly. If $f(b_1) = f(b_2) = 0$ the condition holds trivially. If $f(b_1) > f(b_2) > 0$, then $|f(b_1) - f(b_2)| = b_1 - b_2 + \delta\lambda(A_1 - A_2)/[(1 + A_1)(1 + A_2)] \leq (1 + \delta\lambda\gamma)|b_1 - b_2|$, where $A_k = \gamma(b_k - \lambda)_+$. Finally, if $f(b_1) > 0, f(b_2) = 0$ then w.l.o.g. write $b_1 = \delta\lambda + \epsilon$, $b_2 = \delta\lambda$, for $\epsilon = |b_1 - b_2| > 0$, so that

$$|f(b_1) - f(b_2)| = \epsilon + \delta\lambda \left(1 - \frac{1}{1 + \gamma(\delta\lambda + \epsilon - \lambda)_+} \right). \tag{9}$$

For $\epsilon \leq (1-\delta)\lambda$, (9) becomes ϵ. Otherwise (9) is less than $\epsilon + \delta\lambda\,(\gamma\epsilon/(1 + \gamma\epsilon)) < (1 + \delta\gamma\lambda)\epsilon$.

Note the connection between path continuity and estimator stability: if there are jumps in a solution path as a function of λ, then Lipschitz continuity is impossible. To see this, note that one could choose b_1 and b_2 arbitrarily close but on either side of the discontinuity threshold λ. For this reason it is clear that Lipschitz continuity *does not hold* for $\gamma = \infty$.

6 Implementation via coordinate descent

We use Coordinate descent (CD; e.g., Luenberger and Ye, 2008) to minimize (1) 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. It is conceptually simple, easy to code, and well-suited to optimizations with hot-start initialization. Algorithms of this type have have become dominant 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 (10) 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.1). 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 (1) becomes

$$\underset{\alpha,\beta_1 \dots \beta_p \in \mathbb{R}}{\operatorname{argmin}} \sum_i \frac{v_i}{2}(\alpha + \mathbf{x}_i'\boldsymbol{\beta} - z_i)^2 + n \sum_j \omega_j \lambda |\beta_j|. \tag{10}$$

Our solver iterates between CD on (10) 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. [2]

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 $\hat{\boldsymbol{\eta}} = \hat{\alpha} + \mathbf{X}'\hat{\boldsymbol{\beta}}$.

6.1 Descent convergence

Despite the non-differentiability of $|\beta_j|$ at zero, Tseng (2001) establishes local convergence for CD on (10) 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 this is not always guaranteed, but divergence is rare and we find the algorithm reliable in practice.

6.2 Software

The algorithm is implemented in `c` as part of the `gamlr` package for R. This software is available with documentation on `cran.r-project.org` and versioned source code is at `github.com/mataddy/gamlr`. Usage of `gamlr` mirrors that of its convex penalty ana-

[2] Under high collinearity one can accelerate convergence via a quasi-Newton step (e.g., Lange, 2010) as in Appendix A.2, but given path hot starts this is usually not worth the extra overhead.

logue `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.

7 Related estimators

Many algorithms have been contributed to accompany the literature on concave regularization. For example, aforementioned SCAD and MC+ penalties are implemented for least-squares in `SIS` and `plus` R packages, respectively. This software is more prototypical than the industrial strength tools (like `glmnet`) available for convex regularization.

The recent SparseNet framework of Mazumder et al. (2011) contributes a fast class of coordinate descent routines for generic concave penalized path estimation. The algorithm uses CD to first fit a lasso path and, for each λ on this path, adapts coefficient estimates along a second grid of increasing concavity specifications (analogous to increasing values of our γ). An implementation for Zhang's MC+ penalty is in the `sparsenet` package for R, and it is the best available software we've found. Indeed, we find very similar runtimes for `sparsenet` and roughly equivalent loops of `gamlr` over multiple γ; neither is consistently faster across problems. The advantage of `gamlr` is that each different γ specification can be run in parallel (and you only need use one γ if you wish), whereas the `sparsenet` algorithm moves through a grid of both λ and γ in order to avoid minor modes in the nonconvex penalized objective.

The SparseNet authors argue that it is important in CD to solve exactly the coordinate-wise objective at each update. This will make the solution less sensitive (but not insensitive) to initial values. Thus if the bimodal objective in the right-most panel of Figure 2 represents a coordinate update, one should make sure to find the major mode out near $\beta = 4$. For log penalization, Taddy (2013b) finds these exact solutions by solving a simple quadratic equation. Unfortunately, even this simple exact method becomes prohibitively expensive for p and n above a few thousand (unless, like in SparseNet, you derive good hot-starts from a joint $[\gamma, \lambda]$ grid). The gamma lasso only ever attempts to solve convex optimization problems along the path, so convergence to minor modes is a non-issue.

It turns out that the GL algorithm is quite closely related to the 'local linear approximation' (LLA) routines that are criticized as sub-optimal in Mazumder et al. (2011). Of course, when

the objective itself is uncertain (and needs to be chosen via model selection) the properties of the entire path (e.g. stability) are more important than a single point being optimal under a fixed specification. Indeed, a new literature (Loh and Wainwright, 2013; Wang et al., 2013) holds that, under limited penalty concavity, all local solutions should be within *statistical* precision of the global solution. These papers are technically difficult and the ideas are early in development, but they support the simple point that one should not worry too much about global solutions under a fixed penalty when the optimal specification of that penalty is itself unknown.

7.1 Local linear approximation and one-step estimators

A local linear approximation replaces the concave cost function c in a penalized deviance with its tangent at the current estimate, $c'(\hat{\beta})\beta$. The approximate objective is then a standard lasso (solvable by, say, the CD of Section 6), and one iterates between updating $c'(\hat{\beta})$ and solving the implied L_1 penalized minimization problem. Candes et al. (2008) apply LLA to the log penalty, in which case $c'(\beta) = s\gamma/(1+\gamma|\beta|)$. The method is simple and locally convergent, but it is especially sensitive to initial starting location. Consider approximating the log penalized objectives of Figure 2 with LLA $l(\beta) + c'(0)|\beta|$: the approximation will minimize at $\beta = 0$ until $|l'(\beta)| > c'(0)$, even when the true objective $\beta \neq 0$ local minimum is much lower. Thus in a path estimation algorithm, LLA sticks at zero until it jumps all the way out to near the MLE.

Despite this issue, Zou and Li (2008) present numerical and theoretical evidence that simple LLA does well in practice. For us, the relevant contribution of Zou and Li is their one-step implementation of LLA and its analysis: each GL path segment executes a version of this one-step LLA algorithm. One-step estimation is a general technique (e.g., Bickel, 1975) which amounts to taking as your estimator the first step of an iterative approximation to some objective function. In general, one-step estimators are usually accompanied by results claiming that they can be 'as good' asymptotically as the full-step solution *if* the initial estimates are 'good enough'. Since the efficiency of regularization paths depends upon segment solutions being good starts for the next segment, they form a natural setting for such techniques.

In the context of LLA for the log penalty, one-step estimation implies solving $\hat{\boldsymbol{\beta}}$ to minimize deviance under penalties $\lambda|\beta_j|/(1 + \gamma\hat{\beta}_j^0)$, where $\hat{\boldsymbol{\beta}}^0$ are some initial estimates of the coefficients. For Zou and Li, $\boldsymbol{\beta}^0$ is set to the MLE (or left unspecified), whereas in the gamma

lasso we set it to the previous path segment's estimate. The theoretical framework of one-step estimation, and results from Zou and Li in particular, apply directly to our GL estimator (note that the adaptive lasso of Zou (2006) can also be cast as a one-step LLA). This includes oracle properties if the regularization parameter is appropriately chosen. The Zou and Li result on continuity is more general than our simple statement in Section 5: one-step LLA is continuous if and only if $c'(\beta)$ is continuous for $|\beta| > 0$.

8 Degrees of freedom

An estimation algorithm's degrees of freedom (df) is a key property in determining its tendency to over or under fit. We use it in Section 9 as the basis for selecting the optimal model along a fitted regularization path, so it is a key ingredient for practical application of the GL algorithm.

In an unpenalized linear model, the degrees of freedom is just the number of estimated parameters. More generally, it can be defined as (e.g. Efron et al., 2004) $df = \sigma^{-2} \sum_i \mathrm{cov}(\hat{y}_i, y_i)$ where $\hat{y}_i$ is the estimated prediction rule for observation y_i and σ^2 is residual variance. Given an estimator that is Lipschitz continuous with respect to $\mathbf{y}$, as demonstrated in Section 5 (or in Zou and Li, 2008) for our GL path segments, the SURE framework of Stein (1981) applies and we get a more workable version of this definition, $df = \mathbb{E}\left[\sum_i \partial \hat{y}_i / \partial y_i\right]$.

Consider the case of a single coefficient β estimated via least-squares under L_1 penalization τ. Write gradient at zero $g = -\sum_i x_i y_i$ and curvature $h = \sum_i x_i^2$ and set $\varsigma = -\texttt{sign}(g)$. Then the prediction rule for y given x is $\hat{y} = x(\varsigma/h)(|g| - \tau)_+$ with derivative $\partial \hat{y}/\partial y = x_i^2/h \mathbb{1}_{[|g|<\tau]}$, yielding the SURE degrees of freedom expression $df = \mathbb{E}\left[\mathbb{1}_{[|g|<\tau]}\right]$. For many orthogonal covariates, we get $\hat{y} = \sum_j x_j(\varsigma_j/h_j)(|g_j| - \tau_j)_+$ and the degrees of freedom becomes

$$df = \sum_j \mathbb{E}\left[\mathbb{1}_{[|g_j|<\tau_j]}\right]. \tag{11}$$

Zou et al. (2007) plug sample gradients into (11) to derive $\hat{p} = \sum_j |g_j| < \tau_j = \sum |\hat{\beta}_j| \neq 0$, the number of nonzero coefficients, as an unbiased estimator of model degrees of freedom.

We'll use the same tactic of plugging-in observed gradient. However, for the GL algorithm each τ_j is a function of y. Moreover, unlike g_j, our observed $\hat{\tau}_j$ have been optimized to fit the

data and thus plugging them into (11) would yield a biased (low) estimate of df. Fortunately, interpreting the GL algorithm as an estimator for the Bayesian hierarchical model of Section 3, we do have a guess for each τ_j that is independent from the data: our $\mathrm{Ga}(s, 1/\gamma)$ hyperprior. This implies the *gamma lasso estimator for degrees of freedom*

$$df^t = \sum_j \mathrm{Ga}(|g_j|;\ n\lambda^t/(\gamma\phi), 1/\gamma). \tag{12}$$

The number of unpenalized coefficients (e.g., 1 for α) is always added to this to get total df. For Gaussian regression, where ϕ is unknown, we just use the fitted mean square error $\hat{\phi}^t = \sum_i (y_i - \hat{y}_i)^2/n$. Under (12) as $\gamma \to 0$, $df^t \to \hat{p}$, the lasso df of Zou et al. As $\gamma \to \infty$, df^t becomes p and we assign full model df for all steps of the GL subset selection algorithm.

In the non-orthogonal case, $g_j = g_j(0)$ becomes a function of all of the elements of $\boldsymbol{\beta}$. One option would be to plug-in the null model gradients (i.e. for all zero coefficients), but with any multicollinearity this yields too-large df values (e.g., almost immediately $df \approx p$ in our simulations of Section 10). Instead we use the most recent g_j at which $\hat{\beta}_j^t = 0$, such that the effect of previously in-the-model variables has been remove from our plug-in gradient. This appears to work well in practice, requires no extra computation (since we already need these values in CD), and has the advantage of maintaining $df = \hat{p}$ for $\gamma = 0$.

9 Selection

Penalized estimation does not perform model selection. Rather, its value is as a technique to quickly enumerate, along regularization paths, a set of plausible candidate models. In the simplest setting, the models are indexed by a single penalty size parameter, say λ. This acts as a *squelch*: it suppresses noise to focus on the true input signal. The model selection problem is then to choose λ so that models estimated under that squelch are neither over nor under fit.

9.1 Cross validation

The simplest metric of a model's quality is how well it predicts for data *that you have not observed* drawn from the same (assumed independent across observations) generating process

as the sample $\mathcal{S} = [\mathbf{X}, \mathbf{y}]$. We measure this in terms of deviance under the assumed likelihood: -2 times the fitted log likelihood minus the log likelihood for a saturated model (with $p = n$ free parameters). Since evaluation on actually unobserved data is impossible, it is common to use K-fold cross-validation (CV; see Efron, 2004, for an overview) to estimate predictive performance: split the sample into K evenly-sized disjoint subsets (folds) $\mathcal{S}_k$, and K times fit a regularization path given $\mathcal{S} \setminus \mathcal{S}_k$. Predictions for data in the left-out fold are used to calculate 'out-of- sample' (OOS) deviance along each regularization path. Given CV results, one selects λ according to some criterion and re-fits the model on the entire dataset under this penalty. The usual rules are either $\lambda_{\min}$, the penalty with minimum average OOS error, or $\lambda_{1\text{se}}$, the largest penalty with average OOS error no more than 1 standard error away from that at $\lambda_{\min}$ (to account for downward bias in deviance at $\lambda_{\min}$ as an estimator of true prediction deviance).

CV is super useful, and some form of it features in most applied problems, but it does have some serious weaknesses. As a basic point, K should be bigger than 2-3 (5-10 is typical); if a single path-fit is expensive then doing it K times will be impractical. More subtly, in contemporary Big data problems the data is distributed: it is too large to store on a single machine. Analysts can use conditional independence factorization to work in parallel on data subsets (e.g., Taddy, 2013a), but a computational bottleneck results if you need to communicate across such subsets for OOS experimentation. Thus there are settings where we'll need to replace CV with performance estimates that can be derived from a single model fit.

9.2 Information Criteria

An alternative approach to measuring predictive performance follows Mallows (1973, see also Efron 2004) in estimating true OOS error via the 'Information Criterion',

$$\text{IC}(k) = 2l(\hat{\alpha}, \hat{\boldsymbol{\beta}}) + k \times df \tag{13}$$

where k is a complexity penalty. We are able plug-in the df^t degrees of freedom heuristic from (12) here, and the IC-selected model is that which minimizes the criterion in (13).

The two most common information criteria specifications are the AIC's $k = 2$ (derived from information theoretic arguments in Akaike, 1973) or the BIC's $k = \log(n)$ (derived from

approximation to marginal likelihood for a Bayesian model in Schwarz, 1978). Our experience is that the BIC-selected model works better for OOS prediction in big-p/small-n applications, but it does tend to yield very simple models and the AIC will be preferred if you need to tell a story about weak signals or if $n \gg p$ (e.g., as in Section 11). See Flynn et al. (2013) for discussion of the AIC and its adaptation for selection in high-dimensional penalized regression.

Note that with both IC and CV it is best to use 'stable' models. For IC, Section 8 outlines how the required df can be approximated easily only for models that are Lipschitz continuous. And Breiman (1996) argues that CV does a poor job of evaluating unstable models: it evaluates *average* performance of given λ as a squelch, but this is a good guess at the performance of a single particular fit (which is what we actually have) only if models estimated at this λ are not too variable across samples – that is, if the model is stable. Thus a simple argument for stability is as follows: regularization paths enumerate candidate models that we must choose amongst, and we have good tools for such choice only if the paths are continuous.

9.3 Laplace approximation to the marginal likelihood

An alternative interpretation for information criteria is as Laplace approximation (de Bruijn, 1958) to the negative log of the *marginal likelihood* $\mathrm{p}(\mathbf{y}|\lambda) = \int_{\boldsymbol{\theta}} \mathrm{p}(\mathbf{y}|\boldsymbol{\theta}) d\mathrm{P}_\lambda(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of model parameters and $\mathrm{P}_\lambda(\boldsymbol{\theta})$ is its prior distribution parametrized by λ. For this interpretation of the BIC to apply informally here, $\boldsymbol{\theta}$ is a parametrization of $\mathbf{X}\boldsymbol{\beta}$ in the subspace that results after constraining $\sum_j |\beta_j|$ and $\mathrm{P}_\lambda(\boldsymbol{\theta})$ is the *unit information* Gaussian prior centered at MLE $\hat{\boldsymbol{\theta}}$ with variance $-\partial^2 \log \mathrm{p}(\mathbf{y}|\boldsymbol{\theta})/\partial\theta_j^2|_{\hat{\boldsymbol{\theta}}}$. See Kass and Raftery (1995) and Spiegelhalter et al. (2002) for discussion and references on the connection between Bayes and IC.

A more direct approach to Bayesian model selection would just approximate the marginal likelihood for one of models in Section 3: $\int_{\boldsymbol{\beta}} \mathrm{p}(\mathbf{y}|\boldsymbol{\beta}) d\mathrm{P}_\lambda(\boldsymbol{\beta})$. Unfortunately, Laplace approximation to this integral – replacing the integrand with a normal centered at $\hat{\boldsymbol{\beta}}$ – is inappropriate if the posterior is not differentiable at some $\hat{\beta}_j = 0$. Zhou et al. (2012) propose 'empirical Bayes factors' (EBF) that approximate instead $\int_{\boldsymbol{\beta}_\mathcal{A}} \mathrm{p}(\mathbf{y}|\boldsymbol{\beta}_\mathcal{A}) d\mathrm{P}_\lambda(\boldsymbol{\beta}_\mathcal{A})$, the marginal likelihood over only active-set $\boldsymbol{\beta}_\mathcal{A} = \{\beta_j : \hat{\beta}_j \neq 0\}$.[3] However we find in simulation that this technique,

[3] Yuan and Lin (2005) take a similar approach under a different prior on shared Laplace scale τ, and derive this active set integral as an upper bound on true marginal likelihood.

which is much more computationally intensive, does worse than simple BIC using plug-in df^t.

10 Simulation

Our first example uses data simulated independently from the following p-covariate regression.

$$\begin{aligned} &\mathbf{x} \sim \mathrm{N}\left(\mathbf{0}, \mathbf{\Sigma}\right), \;\; \Sigma_{jk} = 0.9^{|j-k|}, \;\; z_j \overset{ind}{\sim} \mathrm{Bin}(0.5) \;\text{ for }\; j = 1 \ldots p, \\ &y = \mathrm{N}(\mu, \sigma^2) \;\text{ where }\; \mu = (\mathbf{x} * \mathbf{z})'\boldsymbol{\beta}, \;\; \beta_j = \frac{1}{j} \exp\left(-\frac{j}{10}\right), \;\text{ and }\; \sigma = 1.25\mathrm{sd}(\mu). \end{aligned} \tag{14}$$

Here, $\mathrm{sd}(\mu)$ is the sample standard deviation over simulated $\mu_1 \ldots \mu_n$. The true process in (14) is not sparse; the β_j are actually diminishing in absolute size very slowly. However $\mathbf{x}$ is highly correlated across dimensions and it is useless to estimate the coefficients for large j.

Figures 4 through 6 illustrate an example analysis of this data, with $n = 1000$ and $p = 2000$. The paths are initialized at $\lambda^1 = n^{-1} \max\{|g_j(\mathbf{0})|\}_{j=1}^p \approx 0.5$ as in Section 2, and each segment decreases as $\lambda^t = \delta\lambda^{t-1}$ for $\delta = 0.1^{\frac{1}{99}}$, through a grid of 100 values down to $\lambda^{100} \approx 0.05$. Coefficient penalties were divided by the corresponding covariate standard deviation.

The shapes of the regularization paths in Figure 4 show the effect of increasing penalty concavity (via increasing γ), moving from the lasso ($\gamma = 0$) through the larger shoulders of $\gamma = 2$ and on to $\gamma = 10$, where estimates move very quickly to the active-set MLE once a variable is in-the-model. Degrees of freedom, calculated as in (12), are along the top of each plot; equal λ have higher df^t for higher γ since there is less shrinkage of $\hat{\beta}_j \neq 0$. The minimum BIC model is marked upon each path plot with a vertical dashed line; it moves to the right (higher λ) as γ increases, selecting sparser models under more concave penalties.

AIC and BIC values along the paths are plotted in Figure 5. Minimum AIC models are not marked in Figure 4 because it always selected the most complicated model. The BIC curves resemble the 5-fold CV deviance curves in Figure 6, and if one views OOS experiments as a gold-standard then this comparison favors BIC over AIC. In Figure 6, dots mark mean OOS deviance and the bars spread $\pm$ 1 standard error for this statistic. The two CV selection rules, minimum-average $\lambda_{\texttt{min}}$ (left vertical lines) and 1 standard error $\lambda_{\texttt{1se}}$ (right vertical lines), are near each other and select similar models; both are closer to the BIC than it is to the AIC.

A full comparison of selection methods and concavities is shown in Figure 7, which evalu-

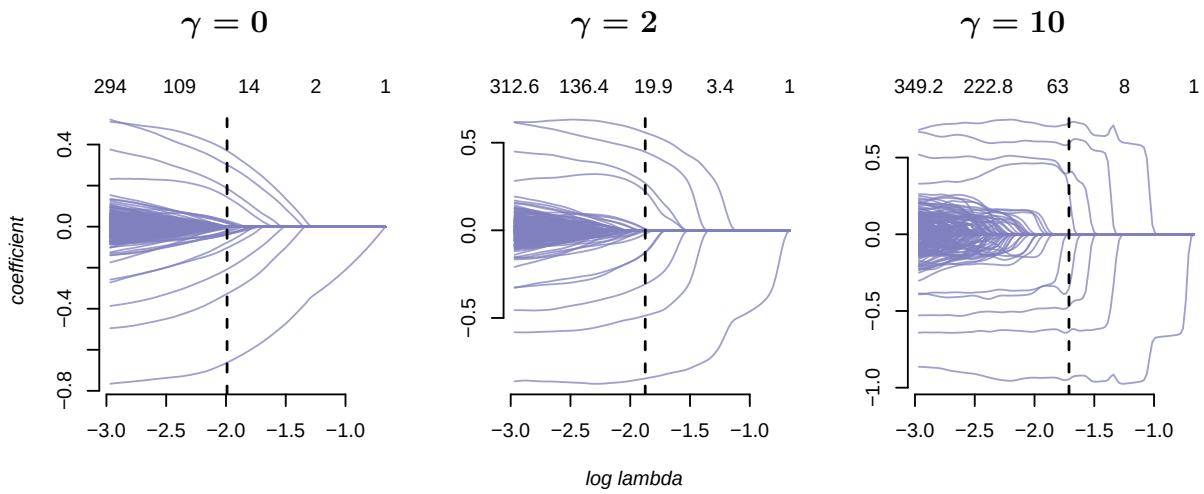

Figure 4: Regularization paths for a single run of our simulated linear regression example, on $\gamma = 0, 2, 10$ from left. Vertical lines mark the BIC selection, and degrees of freedom df^t are along the top.

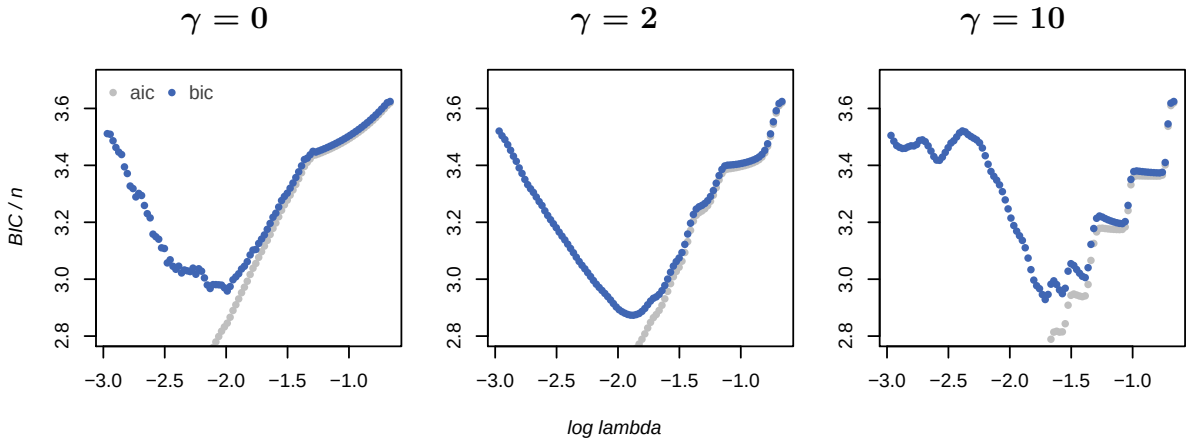

Figure 5: BIC (with AIC in light gray) for our simulated linear regression example, on $\gamma = 0, 2, 10$.

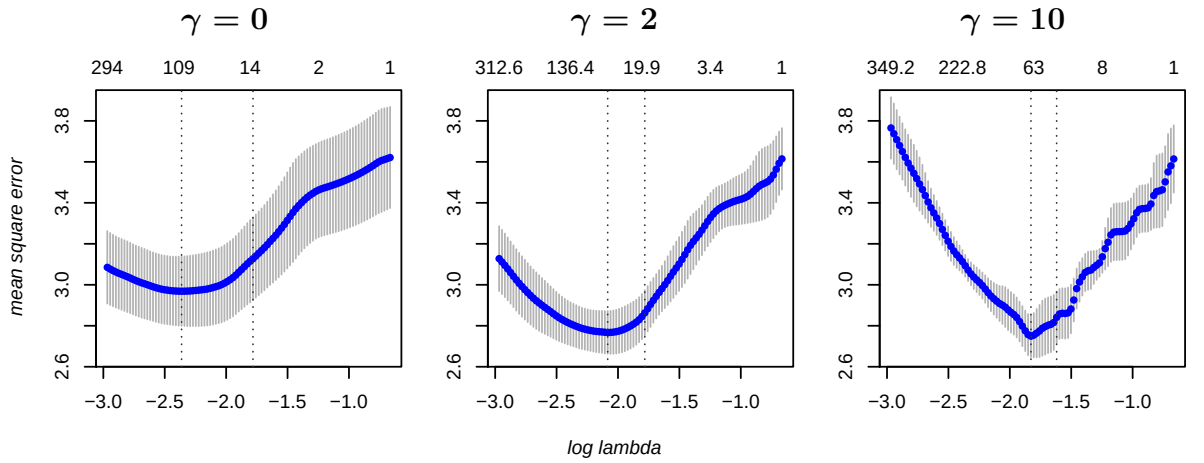

Figure 6: 5-fold CV results for our simulated data example. Points-and-bars show mean OOS deviance $\pm$1se. Minimum and 1se selection rules are marked with dotted lines.

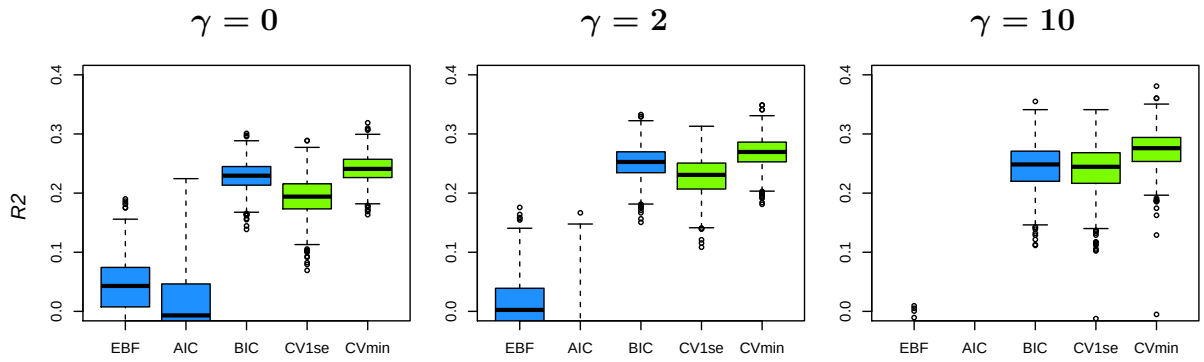

Figure 7: Out-of-sample simulation experiment results.

ated fits on 1000 simulations from (14), with $n = 1000$ and $p = 2000$, and their out-of-sample R^2 (one minus fitted over null deviance) on another 1000 simulations of the same size. For these simulations we again use a grid of 100 segments, but smaller $\lambda^{100} = 0.01\lambda^1$. BIC performs as well as CV, with R^2 falling between $\lambda_{\texttt{min}}$ (better) and $\lambda_{\texttt{1se}}$ (worse) rules. The AIC does far worse here, and actually selects models with majority negative OOS R^2 for $\gamma > 0$. We also show the EBF of Zhou et al. (2012), which does about as well as the AIC.[4]

We do see a slight improvement in performance from $\gamma = 0$ to $\gamma = 2$: 10% gain in average R^2 from 0.225 to 0.25 for BIC selected models (significant given standard errors of about 8e-4). $\gamma = 10$ also has higher average R^2 (0.245 for BIC), but at the expense of more variation around the mean. In some settings the performance gain will be larger and in others it will be smaller. The benefits are probably not worth the trouble of full optimization under a concave penalty, but may be enough to motivate the cheap concavity offered by a gamma lasso with small γ.

11 Hockey players

Our next illustration investigates use of logistic regression to evaluate the performance of hockey players. This is an extension of the 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 for R; there are 64540 goals and 2302 players.

[4]We use posterior MAP for $\hat{\sigma}^2$ in the Zhou et al. formulas. Note that this is a misapplication of the EBF procedure: GL paths only approximate minimization of the log penalized objective for which the EBF was derived.

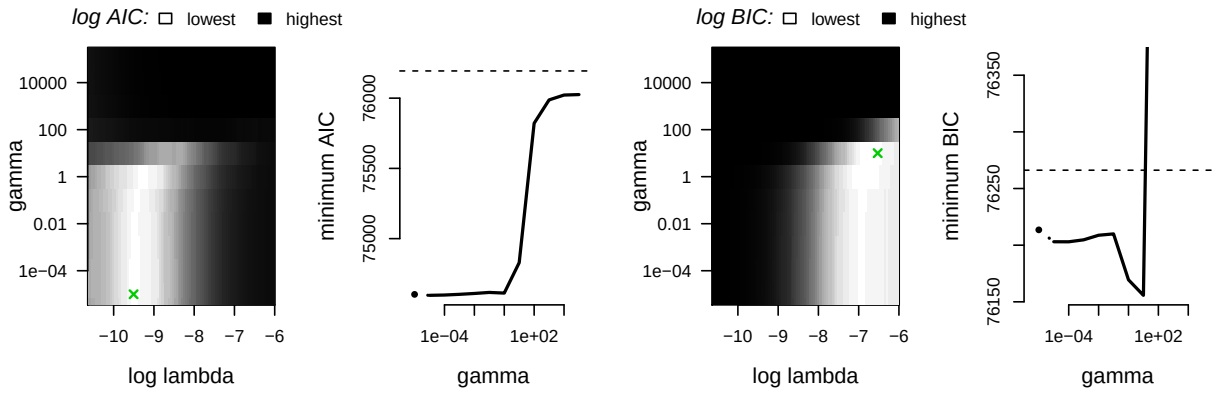

Figure 8: Hockey example AIC and BIC for 100 $\log(\lambda)$ from -6 to -10.6 in models with $\log_{10}\gamma = -5\ldots5$ and lasso $\gamma = 0$. 'X' mark the minima on the images, which have lasso on the bottom edge. Line plots show minimum IC across γ, with a dashed line for null model IC (with α and ϕ only).

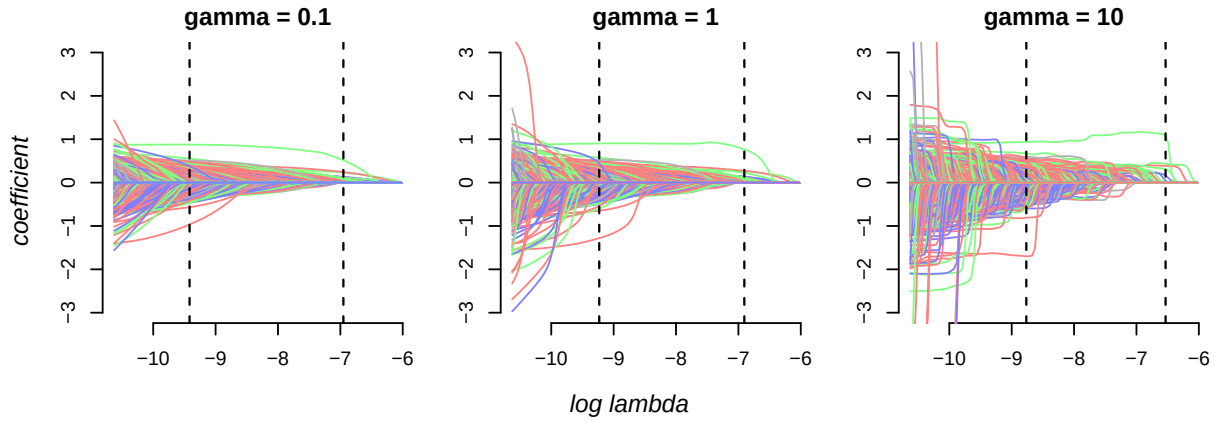

Figure 9: Hockey example regularization paths for $\gamma = 10^{\{-1,0,1\}}$; minimum AIC and BIC are marked with the dashed vertical lines. Positions are shaded grey:goalie, blue:defense, green:center, red:winger.

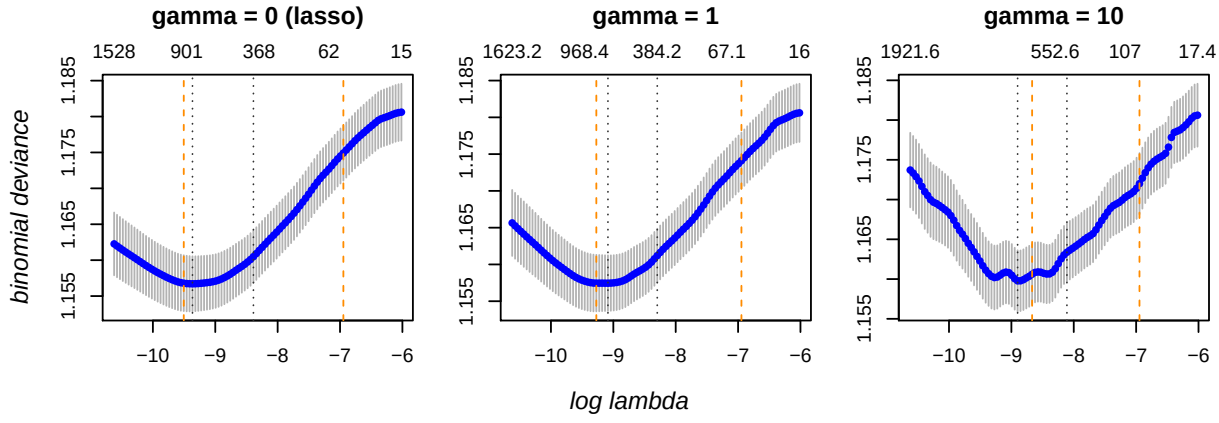

Figure 10: Hockey example 20-fold CV: mean OOS deviance $\pm$1se, with minimum and 1se selection rules marked with black dotted lines. Dashed (orange) lines show AIC and BIC selections.

Our 'regression plus-minus' model for player contribution is then, for goal i,

$$\text{logit}\left[\text{p}(\text{home team scored goal } i)\right] = \alpha + \mathbf{u}'\boldsymbol{\phi} + \mathbf{x}'\boldsymbol{\beta}, \tag{15}$$

where $\mathbf{u}$ is a length-7 vector of indicators for various special-teams scenarios (e.g., a home team power play) and $\mathbf{x}$ is a vector of 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. Thus the β_j coefficients in (15) correspond to the effect of an individual player on the log odds that, given a goal has been scored, the goal was scored by their team. This effect is 'partial' in that it controls for who else was on the ice (and special-teams events)[5], and since β_j do not change from season to season these are 'career effects'.

We estimate gamma lasso paths of $\boldsymbol{\beta}$ for the model in (15), with α and $\boldsymbol{\phi}$ left unpenalized, from $\lambda^1 \approx \exp[-6]$ set as in (3) down to $\lambda^{100} = 0.01\lambda^1$. The penalties were *not* scaled by covariate standard deviation, since this would have favored players with little ice time. The algorithm was run for $\log_{10}\gamma = -5\ldots 5$, plus the $\gamma = 0$ lasso.[6] Joint $[\gamma, \lambda]$ surfaces for AIC and BIC are in Figure 8. The BIC is minimized at higher λ and γ (10 and $e^{-6.5}$) than the AIC (1^{-5} and $e^{-9.5}$), implying more sparsity but with less bias on nonzero β. Figure 8 also contains profiles of BIC or AIC minima across γ: values are low for $\gamma \leq 1$, but jump dramatically at 10 for the AIC and at 100 for the BIC (this is also the threshold start of higher compute times in Figure 3). Figure 9 shows paths on either side of $\gamma = 1$, which appears to exist near a stability threshold: $\gamma \leq 1$ paths are smooth but for $\gamma = 10$ the estimates jump from zero.

The low-BIC models for this data are very sparse. At $\gamma = 1$, the BIC optimal $\log(\lambda) = -6.9$ yields only 21 players with nonzero $\hat{\beta}_j$; these stars (all effects are positive) are shown in Figure 11. This is due in large part to design multicollinearity: since hockey players work on lines and are often matched against lines from the opposing team, it is tough to isolate the performance of individuals. With concave penalties imposing little bias on large signals, a few stars are assigned the aggregate effect of their team-mates and common opponents.

The AIC favors denser models with more bias. The minimum-AIC lasso ($\gamma = 0$) model

[5]Since goalies stay on-ice for long periods (often entire games) their β_j act as a control for overall team quality. Thus a skater (non-goalie) β_j only need be nonzero if that player is significantly above or below the team average.
[6]On this data, $\gamma = \infty$ subset selection yields perfect separation and infinite likelihood.

identifies 904 nonzero player effects; the 20 largest (in absolute value) are in Figure 12. Such estimates will be useful to hockey analysts who wish to evaluate more than just a few key players. The increased bias also conforms to our prior expectation that single players cannot have an overly large effect on the game. Indeed, the 20-fold CV results in Figure 10 support lower λ: AIC selections are near the minimum OOS deviance, and BIC selections are well to the right of the 1se OOS deviance rule. Comparing these results to those of Section 10, where the BIC was clearly superior, illustrates that no single model selection rule is uniformly best. Rather, AIC and BIC provide informal bounds on model complexity; use your knowledge of the application-at-hand (and OOS experiments if possible) to make a final decision.

PETER_FORSBERG	MARIAN_HOSSA	PAVEL_DATSYUK	HENRIK_SEDIN
0.768	0.276	0.268	0.221
NICKLAS_LIDSTROM	ZDENO_CHARA	SIDNEY_CROSBY	DANIEL_ALFREDSSON
0.137	0.134	0.115	0.114
DAN_BOYLE	NATHAN_HORTON	JOE_THORNTON	JONATHAN_TOEWS
0.114	0.097	0.092	0.085
ALEX_OVECHKIN	RYAN_GETZLAF	CHRIS_KUNITZ	JASON_ARNOTT
0.075	0.066	0.052	0.039
ALEX_TANGUAY	VINCENT_LECAVALIER	MARIAN_GABORIK	ZACH_PARISE
0.036	0.033	0.019	0.015
ROBERTO_LUONGO			
0.002			

Figure 11: All 21 non-zero player effects in the $\gamma = 1$ model at BIC optimal $\log(\lambda) = -6.9$.

NINO_NIEDERREITER	PETER_FORSBERG	RAITIS_IVANANS	BRAD_MARCHAND
-0.955	0.873	-0.568	0.560
PATRICK_ROY	MARTIN_GELINAS	ERIC_FEHR	THOMAS_POCK
0.511	0.488	0.486	-0.476
KENT_MANDERVILLE	JONATHAN_TOEWS	MARIAN_HOSSA	STEVEN_GOERTZEN
-0.471	0.466	0.453	-0.442
MATHIEU_BIRON	SIDNEY_CROSBY	RYAN_HOLLWEG	TIM_TAYLOR
-0.439	0.433	-0.430	-0.419
SCOTT_LACHANCE	PAVEL_DATSYUK	ALEXANDER_SEMIN	DALTON_PROUT
-0.418	0.416	0.408	0.407

Figure 12: 20 largest player effects (of 904 nonzero) in the lasso model at AIC optimal $\log(\lambda) = -9.5$.

12 Discussion

We have found the gamma lasso algorithm to be useful and effective in application. Despite an abundance of theoretical and empirical evidence showing that concave penalties are superior to L_1 costs (for example, see the references in Section 7), our experience is that it is tough to do *much* better than the simple lasso. And it is easy to do worse. The stable and fast gamma lasso algorithm provides an appealing middle ground.

Despite our advocacy of the GL algorithm, this article was designed to be more of a review than an outline of new methods. Our discussion finds that the GL is very closely related to many different techniques in the dense literature on concave regularization: it approximates MAP estimation of a Bayesian model, or deviance minimization under log penalties, and is really 'just' a pathwise implementation of one-step LLA (or the adaptive lasso). A central point of our review is that it is important to understand the entire regularization path (especially its continuity). This is especially true under concave penalties, where due to multi-modality the solutions depend (through hot-starts) on the route the path takes. The point is made even more explicit in GL estimation, where the size of the gap $\lambda^{t-1} - \lambda^t$ changes the solution at λ^t.

Finally, any regularization path algorithm merely builds a set of candidate models. Selection amongst these models is an essential extra step. In our case, we are selecting the 'stopping point' t and corresponding $\hat{\boldsymbol{\beta}}^t$. Heuristic degrees of freedom for GL paths, and resulting AIC/BIC metrics, allow such selection through an intuitive extension of lasso methods. CV is great, and easiest to explain to non-statisticians, but it nice to have a fast analytic alternative.

A Appendix

A.1 Gradient and curvature

The simplified negative log likelihood objective in Gaussian regression is $l(\alpha, \boldsymbol{\beta}) = 0.5 \sum_i (y_i - \eta_i)^2$ with gradient and coordinate curvature

$$g_j(\boldsymbol{\beta}) = \frac{\partial l}{\partial \beta_j} = -\sum_i x_{ij}(y_i - \eta_i), \qquad h_j(\boldsymbol{\beta}) = \frac{\partial^2 l}{\partial \beta_j^2} = \sum_i x_{ij}^2.$$

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 $l(\alpha, \boldsymbol{\beta}) = \sum_i -y_i\eta_i + \log(1 + \exp[\eta_i])$ and

$$g_j(\boldsymbol{\beta}) = \frac{\partial l}{\partial \beta_j} = -\sum_i x_{ij}(y_i - q_i), \qquad h_j(\boldsymbol{\beta}) = \frac{\partial^2 l}{\partial \beta_j^2} = \sum_i x_{ij}^2 q_i(1 - q_i).$$

A.2 Quasi-Newton acceleration

Acceleration is applied to $\boldsymbol{\theta} = [\alpha, \boldsymbol{\beta}]$, the set of both penalized and unpenalized parameters. This move is accepted only if it leads to a decrease in the objective, such that there is no effect on algorithm convergence. 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 $\partial M/\partial\hat{\theta}_l \approx \mathrm{v}_l/\mathrm{u}_l$. An approximate Newton-Raphson step to solve for the root of $\hat{\boldsymbol{\theta}} - M(\hat{\boldsymbol{\theta}})$ updates each coordinate

$$\hat{\theta}_l \leftarrow \hat{\theta}_l^{(-1)} - (\hat{\theta}_l^{(-1)} - \hat{\theta}_l^{(0)})/(1 - \mathrm{v}_l/\mathrm{u}_l)$$

which can be re-written as $\hat{\theta}_l = (1 - \mathrm{w}_l)\hat{\theta}_l^{(-1)} + \mathrm{w}_l\hat{\theta}_l^{(0)}$ where $\mathrm{w}_l = \mathrm{u}_l/(\mathrm{u}_l - \mathrm{v}_l)$.

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*.

Armagan, A., D. B. Dunson, J. Lee, W. U. Bajwa, and N. Strawn (2013). Posterior consistency in linear models under shrinkage priors. *To appear in Biometrika*.

Belloni, A., V. Chernozhukov, and C. Hansen (2012). Inference on treatment effects after selection amongst high-dimensional controls. MIT Department of Economics Working Paper No. 12-13.

Bickel, P. J. (1975). One-step Huber estimates in the linear model. *Journal of the American Statistical Association 70*, 428—434.

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

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)*.

de Bruijn, N. G. (1958). *Asymptotic Methods in Analysis*. Dover.

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.

Flynn, C., H. Clifford, and S. Jefferey (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.

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.

Kass, R. E. and A. E. Raftery (1995). Bayes factors. *Journal of the American Statistical Association 90*, 773795.

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

Loh, P.-L. and M. J. Wainwright (2013). Regularized m-estimators with nonconvexity: Statistical and algorithmic theory for local optima. arXiv:1305.2436.

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.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics 6*, 461464.

Spiegelhalter, D. J., N. G. Best, B. P. Carlin, and A. Van Der Linde (2002). Bayesian measures of model complexity and fit. *Journal of the Royal Statistical Society: Series B (Statistical Methodology) 64*, 583–639.

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.

Wang, Z., H. Liu, and T. Zhang (2013). Optimal computational and statistical rates of convergence for sparse nonconvex learning problems. arXiv:1306.49060.

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

Yuan, M. and Y. Lin (2005). Efficient empirical bayes variable selection and estimation in linear models. *Journal of the American Statistical Association 100*, 1215–1225.

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

Zhou, H., A. Armagan, and D. B. Dunson (2012). Path following and empirical Bayes model selection for sparse regression. arXiv:1201.3582.

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

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.