Effective Degrees of Freedom: A Flawed Metaphor

Lucas Janson, Will Fithian, Trevor Hastie

December 30, 2013

Abstract

To most applied statisticians, a fitting procedure's degrees of freedom is synonymous with its model complexity, or its capacity for overfitting to data. In particular, it is often used to parameterize the bias-variance tradeoff in model selection. We argue that, contrary to folk intuition, model complexity and degrees of freedom are *not* synonymous and may correspond very poorly. We exhibit various examples of fitting procedures for which degrees of freedom is not monotonic in the model complexity parameter, and can exceed the total dimension of the largest model. Even in very simple settings, the degrees of freedom can exceed the dimension of the ambient space by an arbitrarily large amount.

1 Introduction

To motivate our inquiry, consider estimating a no-intercept linear regression model with design matrix $\boldsymbol{X} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$ and $\boldsymbol{y} \sim N(\boldsymbol{\mu}, I)$, with $\boldsymbol{\mu} \in \mathbb{R}^2$. Suppose further that, in order to obtain a more parsimonious model than the full bivariate regression, we instead estimate the best fitting of the two univariate models (in other words, best subsets regression with model size $k = 1$). What is the effective degrees of freedom (DF) of the fitted model in this seemingly innocuous setting?

A simple, intuitive, and wrong argument predicts that the DF lies somewhere between 1 and 2. We expect it to be greater than 1, since we use the data to select the better of two one-dimensional models. However, we have only two free parameters at our disposal, with a rather severe constraint on the values they are allowed to take. Our procedure is strictly less complex than the saturated model with two parameters that fits the data as hard as possible.

Figure 1 shows the effective DF for this model, plotted as a function of $\boldsymbol{\mu} \in \mathbb{R}^2$, the expectation of the response $\boldsymbol{y}$. As expected, $\mathrm{DF}(\boldsymbol{\mu}) \geq 1$, with near-equality when $\boldsymbol{\mu}$ is close to one of the coordinate axes but far from the other. Perhaps surprisingly, however, $\mathrm{DF}(\boldsymbol{\mu})$ can exceed 2, approaching 7 in the corners of the plot. Indeed, the plot suggests (and we will later confirm) that

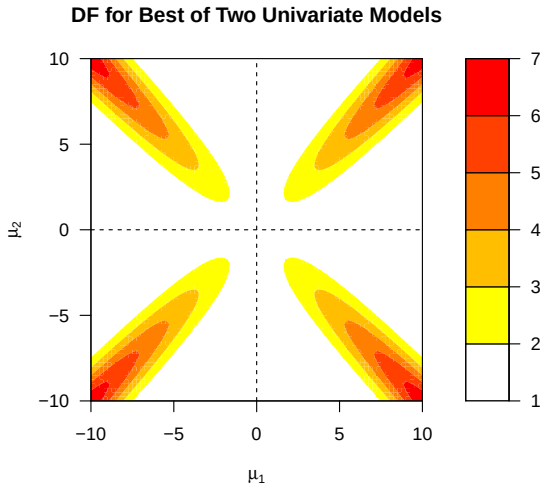

Figure 1: The DF the best univariate submodel, for identity design matrix with $n = p = 2$. The DF is plotted here as a function of μ. Contrary to what one might naively expect, the DF can significantly exceed 2, the DF for the full model.

$\mathrm{DF}(\boldsymbol{\mu})$ can grow arbitrarily large as $\boldsymbol{\mu}$ moves farther diagonally from the origin.

To understand why our intuition should lead us astray here, we must first review how the DF is defined for a general fitting procedure, and what classical concept that definition is meant to generalize.

1.1 Degrees of Freedom in Classical Statistics

The original meaning of degrees of freedom, the number of dimensions in which a random vector may vary, plays a central role in classical statistics. Consider for example an ordinary linear regression with n-dimensional response vector $\boldsymbol{y}$ and $n \times p$ predictor matrix $\boldsymbol{X}$ with full column rank. The fitted response $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ is the orthogonal projection of $\boldsymbol{y}$ onto the p-dimensional column space of $\boldsymbol{X}$, and the residual $\boldsymbol{r} = \boldsymbol{y} - \hat{\boldsymbol{y}}$ is the projection onto its orthogonal complement, whose dimension is $n - p$. We say this linear model has p "model degrees of freedom" (or just "degrees of freedom"), with $n - p$ "residual degrees of freedom."

If the error variance is σ^2, then $\boldsymbol{r}$ is "pinned down" to have zero projection in p directions, and is free to vary, with variance σ^2, in the remaining $n - p$ orthogonal directions. In particular, if the model is correct ($\mathbb{E}\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta}$), then the residual sum of squares (RSS) has distribution

$$\mathrm{RSS} = \|\boldsymbol{r}\|_2^2 \sim \sigma^2 \cdot \chi^2_{n-p}. \tag{1}$$

It follows that $\mathbb{E}\|\boldsymbol{r}\|_2^2 = \sigma^2(n-p)$, leading to the unbiased variance estimate $\hat{\sigma}^2 = \frac{1}{n-p}\|\boldsymbol{r}\|_2^2$. Most inference procedures for linear models, such as t-tests and F-tests, are similarly based on analyzing lengths of n-variate Gaussian random vectors after projecting onto appropriate linear subspaces.

In linear regression, the model degrees of freedom (henceforth DF) serves to quantify multiple related properties of the fitting procedure. The DF coincides with the number of non-redundant free parameters in the model, and thus constitutes a natural measure of model complexity or overfitting. In addition, the total variance of the fitted response $\hat{\boldsymbol{y}}$ is exactly $\sigma^2 p$, which depends only on the number of linearly independent predictors and not on their size or correlation with each other.

The DF also quantifies *optimism* of the residual sum of squares as an estimate of out-of-sample prediction error. In linear regression, one can easily show that the RSS understates mean squared prediction error by $2\sigma^2 p$ on average. Mallows (1973) proposed exploiting this identity as a means of model selection, by computing $\mathrm{RSS} + 2\sigma^2 p$, an unbiased estimate of prediction error, for several models, and selecting the model with the smallest estimated test error. Thus, the DF of each model contributes a sort of penalty for how hard that model is fitted to the data.

1.2 "Effective" or "Equivalent" Degrees of Freedom

For more general fitting procedures such as smoothing splines, generalized additive models, or ridge regression, the number of free parameters is often either undefined or an inappropriate measure of model complexity. Most such methods feature some tuning parameter modulating the fit's complexity, but it is not clear a priori how to compare e.g. a Lasso with Lagrange parameter $\lambda = 3$ to a local regression with window width 0.5. When comparing different methods, or the same method with different tuning parameters, it can be quite useful to have some measure of complexity with a consistent meaning across a diverse range of algorithms. To this end, various authors have proposed alternative, more general definitions of a method's "effective degrees of freedom," or "equivalent degrees of freedom" (see Buja et al. (1989) and references therein).

If the method is linear — that is, if $\hat{\boldsymbol{y}} = \boldsymbol{H}\boldsymbol{y}$ for some "hat matrix" $\boldsymbol{H}$ that is not a function of $\boldsymbol{y}$ — then the trace of $\boldsymbol{H}$ serves as a natural generalization. For linear regression $\boldsymbol{H}$ is a p-dimensional projection, so $\mathrm{tr}(\boldsymbol{H}) = p$, coinciding with the original definition. Intuitively, when $\boldsymbol{H}$ is not a projection, $\mathrm{tr}(\boldsymbol{H})$ accumulates fractional degrees of freedom for directions of $\boldsymbol{y}$ that are shrunk, but not entirely eliminated, in computing $\hat{\boldsymbol{y}}$.

For nonlinear methods, further generalization is necessary. The most popular definition, due to Efron (1986) and given in Equation (5), defines DF in terms of the optimism of RSS as an estimate of test error, and applies to any fitting method.

Measuring or estimating optimism is a worthy goal in and of itself. But to justify our intuition that the DF offers a consistent way to quantify model

complexity, a bare requirement is that the DF be monotone in model complexity when considering a fixed method.

The term "model complexity" is itself rather metaphorical when describing arbitrary fitting algorithms, but has a concrete meaning for methods that minimize RSS subject to the fit $\hat{\boldsymbol{y}}$ belonging to a closed constraint set $\mathcal{M}$ (a "model"). Commonly, some tuning parameter γ selects one of a nested set of models:

$$\hat{\boldsymbol{y}}^{(\gamma)} = \underset{\boldsymbol{z} \in \mathcal{M}_\gamma}{\arg\min} \|\boldsymbol{y} - \boldsymbol{z}\|_2^2, \quad \text{with } \mathcal{M}_{\gamma_1} \subseteq \mathcal{M}_{\gamma_2} \subseteq \mathbb{R}^n \text{ if } \gamma_1 \leq \gamma_2 \tag{2}$$

Examples include the Lasso (Tibshirani, 1996) and ridge regression (Hoerl, 1962) in their constraint formulation, as well as best subsets regression (BSR). The model $\mathcal{M}_k$ for BSR with k variables is a union of k-dimensional subspaces.

Because larger models "fit harder" to the data, one naturally expects DF to be monotone with respect to model inclusion. For example, one might imagine a plot of DF versus k for BSR to look like Figure 2 when the full model has $p = 10$ predictors: monotone and sandwiched between k and p.

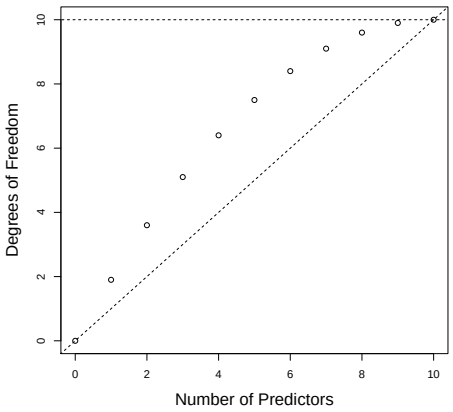

Figure 2: This plot follows the usual intuition for degrees of freedom for best subset regression when the full model has 10 predictors.

However, as we have already seen in Figure 1, monotonicity is far from guaranteed even in very simple examples. Kaufman and Rosset (2013) independently report on non-monotonicity of the degrees of freedom for Lasso and ridge regression. We additionally consider fitting methods with non-convex constraint sets, proving that for any method that projects $\boldsymbol{y}$ onto a non-convex set $\mathcal{M} \subseteq \mathbb{R}^n$, the degrees of freedom can be arbitrarily large.

2 Preliminaries

Although the phenomenon discussed in this paper occurs much more generally, we will focus on the following linear model,

$$\boldsymbol{y} = \boldsymbol{X}\boldsymbol{\beta} + \boldsymbol{\varepsilon}, \tag{3}$$

where all variables are vector-valued: $\boldsymbol{\beta} \in \mathbb{R}^p$ and $\boldsymbol{y}, \boldsymbol{\varepsilon} \in \mathbb{R}^n$. The ε_i are taken to be independently and identically distributed (*i.i.d.*). Note $\boldsymbol{X} \in \mathbb{R}^{n \times p}$ has rows $\boldsymbol{x}_i^T$, the observation vectors.

In order to fit such a model to a dataset $(\boldsymbol{y}, \boldsymbol{X})$, we consider fitting techniques with some tuning parameter (discrete or continuous) that can be used to vary the model from less to more constrained. In BSR, the tuning parameter k determines how many predictor variables are retained in the model, and we will denote BSR with tuning parameter k as BSR_k. For a general fitting technique FIT, we will use the notation FIT_k for FIT with tuning parameter k, and $\hat{\boldsymbol{y}}^{(FIT_k)}$ and $\hat{\boldsymbol{\beta}}^{(FIT_k)}$ for the fitted response and parameter vectors, respectively, produced by FIT_k.

As mentioned in the introduction, a general formula for DF can be motivated by the following relationship between expected prediction error (EPE) and RSS for ordinary least squares (OLS) (Mallows, 1973):

$$\text{EPE} = \mathbb{E}[\text{RSS}] + 2\sigma^2 p. \tag{4}$$

Analogously, once a fitting technique (and tuning parameter) FIT_k is chosen for fixed data $(\boldsymbol{y}, \boldsymbol{X})$, DF is defined by the following identity:

$$\mathbb{E}\left[\sum_{i=1}^{n}(y_i^* - \hat{y}_i^{(FIT_k)})^2\right] = \mathbb{E}\left[\sum_{i=1}^{n}(y_i - \hat{y}_i^{(FIT_k)})^2\right] + 2\sigma^2 \cdot \text{DF}(\boldsymbol{\beta}, \boldsymbol{X}, \text{FIT}_k), \tag{5}$$

where σ^2 is the variance of the ε_i (which we assume exists), and y_i^* is a new independent copy of y_i at $\boldsymbol{x}_i$. Thus DF is defined as a measure of the optimism of RSS. This definition in turn leads to a simple closed form expression for DF under very general conditions, as shown by the following theorem.

Theorem 1 (Efron (1986))**.** *For $i \in \{1, \ldots, n\}$, let $y_i = \mu_i + \varepsilon_i$, where the μ_i are nonrandom and the ε_i have mean zero and finite variance. Let $\hat{y}_i$, $i \in \{1, \ldots, n\}$ denote estimates of μ_i from some fitting technique based on a fixed realization of the y_i, and let y_i^*, $i \in \{1, \ldots, n\}$ be independent of and identically distributed as the y_i. Then*

$$\mathbb{E}\left[\sum_{i=1}^{n}(y_i^* - \hat{y}_i)^2\right] - \mathbb{E}\left[\sum_{i=1}^{n}(y_i - \hat{y}_i)^2\right] = 2\sum_{i=1}^{n}\text{Cov}(y_i, \hat{y}_i) \tag{6}$$

Proof. For $i \in \{1, \ldots, n\}$,

$$\begin{aligned}
\mathbb{E}\big[(y_i^* - \hat{y}_i)^2\big] &= \mathbb{E}\big[(y_i^* - \mu_i + \mu_i - \hat{y}_i)^2\big] \\
&= \mathbb{E}\big[(y_i^* - \mu_i)^2\big] + 2\mathbb{E}\big[(y_i^* - \mu_i)(\mu_i - \hat{y}_i)\big] + \mathbb{E}\big[(\mu_i - \hat{y}_i)^2\big] \qquad (7) \\
&= \text{Var}(\varepsilon_i) + \mathbb{E}\big[(\mu_i - \hat{y}_i)^2\big],
\end{aligned}$$

where the middle term in the second line equals zero because y_i^* is independent of all the y_j and thus also of $\hat{y}_i$, and because $\mathbb{E}\big[y_i^* - \mu_i\big] = 0$. Furthermore,

$$\begin{aligned}\mathbb{E}\big[(y_i - \hat{y}_i)^2\big] &= \mathbb{E}\big[(y_i - \mu_i + \mu_i - \hat{y}_i)^2\big] \\ &= \mathbb{E}\big[(y_i - \mu_i)^2\big] + 2\mathbb{E}\big[(y_i - \mu_i)(\mu_i - \hat{y}_i)\big] + \mathbb{E}\big[(\mu_i - \hat{y}_i)^2\big] \qquad (8) \\ &= \mathrm{Var}(\varepsilon_i) + \mathbb{E}\big[(\mu_i - \hat{y}_i)^2\big] - 2\,\mathrm{Cov}(y_i, \hat{y}_i).\end{aligned}$$

Finally, subtracting Equation (8) from Equation (7) and summing over $i \in \{1, \ldots, n\}$ gives the desired result. □

Remark 1. *For i.i.d. errors with finite variance σ^2, Theorem 1 implies that,*

$$DF(\boldsymbol{\beta}, \boldsymbol{X}, FIT_k) = \frac{1}{\sigma^2}\,\mathrm{tr}\left(\,\mathrm{Cov}(\boldsymbol{y}, \hat{\boldsymbol{y}}^{(FIT_k)})\right) = \frac{1}{\sigma^2}\sum_{i=1}^{n}\mathrm{Cov}\left(y_i, \hat{y}_i^{(FIT_k)}\right). \qquad (9)$$

Note also that when FIT_k is a linear fitting method with hat matrix $\boldsymbol{H}$, Equation (9) reduces to

$$DF(\boldsymbol{\beta}, \boldsymbol{X}, FIT_k) = \mathrm{tr}(\boldsymbol{H}). \qquad (10)$$

3 Additional Examples

For each of the following examples, a model-fitting technique, mean vector, and noise process are chosen. Then the DF is estimated by Monte Carlo simulation. The details of this estimation process, along with all R code used, are provided in the appendix.

3.1 Best Subset Regression Example

Our first motivating example is meant to mimic a realistic application. This example is a linear model with $n = 50$ observations on $p = 15$ variables and a standard Gaussian noise process. The design matrix $\boldsymbol{X}$ is a $n \times p$ matrix of *i.i.d.* standard Gaussian noise. The mean vector $\boldsymbol{\mu}$ (or equivalently, the coefficient vector $\boldsymbol{\beta}$) is generated by initially setting the coefficient vector to the vector of ones, and then standardizing $\boldsymbol{X}\boldsymbol{\beta}$ to have mean zero and standard deviation seven. We generated a few $(\boldsymbol{X}, \boldsymbol{\beta})$ pairs before we discovered one with substantially non-monotonic DF, but then measured the DF for that $(\boldsymbol{X}, \boldsymbol{\beta})$ to high precision via Monte Carlo.

The left plot of Figure 3 shows the plot of Monte Carlo estimated DF versus subset size. The right plot of Figure 3 shows the same plot for forward stepwise regression (FSR) applied to the same data. Although FSR isn't quite a constrained least-squares fitting method, it is a popular method whose complexity is clearly increasing in k. Both plots show that the DF for a number of subset sizes is greater than 15, and we know from standard OLS theory that the DF for subset size 15 is exactly 15 in both cases.

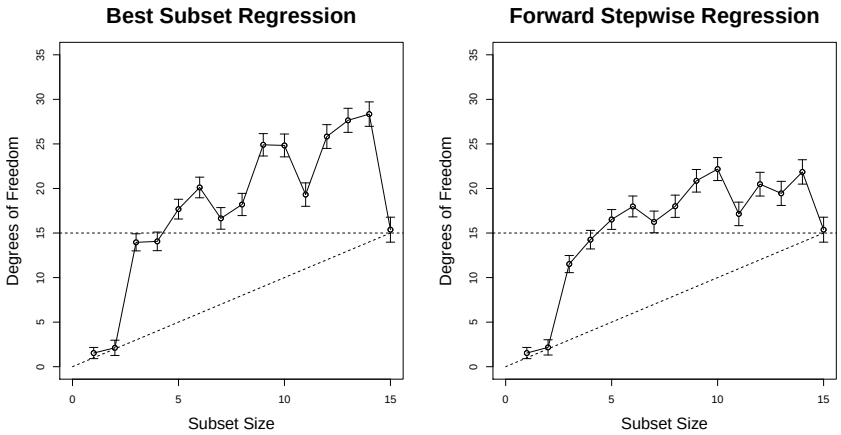

Figure 3: Monte Carlo estimated degrees of freedom versus subset size in the best subset regression (left) and forward stepwise regression (right) examples. Estimates are shown plus or minus two (Monte Carlo) standard errors.

3.2 Lasso Example

This example[1] is small for clarity, but is representative of a much broader class of datasets. The key idea is that in the full Lasso path, variables can be both added and removed as the ℓ_1 norm increases. For the Lasso, we have the following results relating DF to the number of nonzero coefficients in the fitted model (NNZ):

- $\mathrm{DF} = \mathbb{E}[\mathrm{NNZ}]$ for the penalized form of the Lasso (Zou et al., 2007; Tibshirani and Taylor, 2012).
- $\mathrm{DF} = \mathbb{E}[\mathrm{NNZ}] - 1$ for the constrained form of the Lasso (Kato, 2009).

Thus by constructing data for which a variable is consistently removed around the same penalty or constraint value, we can ensure that the DF drops near the same point as well.

The linear model setup is

$$\boldsymbol{X} = \begin{bmatrix} 0 & 1 \\ 2 & -5 \end{bmatrix}, \quad \boldsymbol{\beta} = \begin{pmatrix} -6 \\ -1 \end{pmatrix}, \quad \boldsymbol{\varepsilon} \overset{i.i.d.}{\sim} N(0, 0.03^2).$$

The model was fit using the contrained version of the Lasso over a range of ℓ_1 constraint values. Figure 4 shows the Monte Carlo estimated DF versus ℓ_1 constraint, along with an aligned plot of the solution path. The figure shows a temporary reversal of DF monotonicity with respect to ℓ_1 constraint at around 3.3. Here, a set corresponding to a smaller ℓ_1 constraint is always a (strict) subset of any set corresponding to a larger ℓ_1 constraint.

[1] We note that Kaufman and Rosset (2013) independently came up with a similar example.

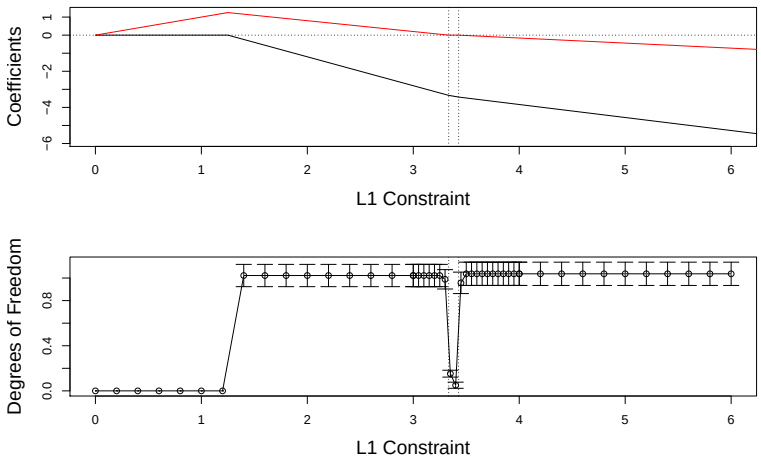

Figure 4: Top: Solution path for Lasso example. Bottom: Monte Carlo estimated degrees of freedom versus ℓ_1 norm in the Lasso example. Estimates are shown plus or minus two (Monte Carlo) standard errors. Dotted lines span the ℓ_1 norm values for which $\beta_2 = 0$.

3.3 Unbounded Degrees of Freedom Example

Finally, we show the motivating example from the introduction, whose result is quite striking. The setup is again a linear model, but with $n = p = 2$. The design matrix $\boldsymbol{X} = A \cdot \boldsymbol{I}$, where A is a scalar and $\boldsymbol{I}$ is the (2×2) identity matrix. The coefficient vector is just $(1, 1)^T$ making the mean vector $(A, A)^T$, and the noise is *i.i.d.* standard Gaussian. Considering BSR_1, it is clear that the best univariate model just chooses the larger of the two response variables, for each realization. Simulating 10^5 times with $A = 10^4$, we get a Monte Carlo estimate of the DF of BSR_1 to be about 5630, with a standard error of about 26. Playing with the value of A reveals that for large A, DF is approximately linearly increasing in A. This suggests that not only is the DF greater than $n = 2$, but can be made unbounded without changing n or the structure of $\boldsymbol{X}$. Figure 5 shows what is going on for a few points (and a smaller value of A for visual clarity). The variance of the y_i (the black points) is far smaller than that of the $\hat{y}_i$ (the projections of the black dots onto the blue constraint set). Therefore, since the correlation between the y_i and the $\hat{y}_i$ is around 0.5, $\sum_{i=1}^{n} \text{Cov}(y_i, \hat{y}_i)$ is also much larger than the variance of the y_i, and the large DF can be infered from Equation (9).

In this toy example, we can actually compute the exact DF using Equa-

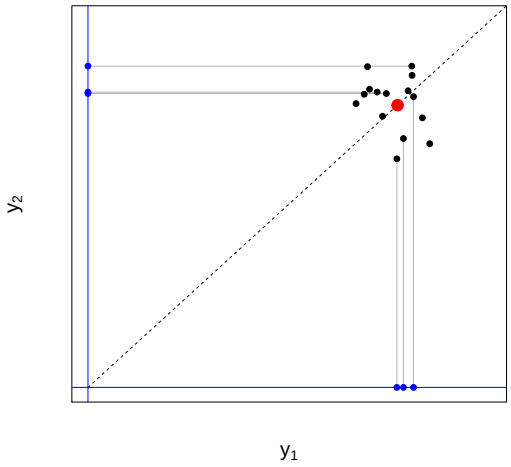

Figure 5: Sketch of the unbounded DF example (not to scale with the A value used in the text). The true mean vector is in red, the constraint region in blue, and some data points are shown in black. The grey lines connect some of the data points to their projections onto the constraint set, in blue. Note that for $i \in \{1, 2\}$, $\hat{\boldsymbol{y}}_i$ is either 0 or approximately A depending on small changes in $\boldsymbol{y}$.

tion (5),

$$\begin{aligned}
\mathrm{DF}(\boldsymbol{\beta}, \boldsymbol{X}, \mathrm{BSR}_1) &= \frac{1}{2}\Big(\mathbb{E}\left[\sum_{i=1}^{n}(y_i^* - \hat{y}_i^{(BSR_1)})^2\right] - \mathbb{E}\left[\sum_{i=1}^{n}(y_i - \hat{y}_i^{(BSR_1)})^2\right]\Big) \\
&= \frac{1}{2}\Big(\big(2 + A^2 + \mathrm{Var}(\max(\varepsilon_1, \varepsilon_2)) + \mathbb{E}\big[\max(\varepsilon_1, \varepsilon_2)\big]^2\big) \\
&\qquad - \big(A^2 + \mathrm{Var}(\max(\varepsilon_1, \varepsilon_2)) + \mathbb{E}\big[\max(\varepsilon_1, \varepsilon_2)\big]^2 \\
&\qquad\qquad - 2A \cdot \mathbb{E}\big[\max(\varepsilon_1, \varepsilon_2)\big]\big)\Big) \\
&= 1 + A \cdot \mathbb{E}\big[\max(\varepsilon_1, \varepsilon_2)\big],
\end{aligned} \tag{11}$$

where $\mathbb{E}\big[\max(\varepsilon_1, \varepsilon_2)\big] \approx 0.5642$, in good agreement with the simulated result.

4 Geometric Picture

The explanation for the non-monotonicity of DF can be best understood by thinking about regression problems as constrained optimizations. We will see that the reversal in DF monotonicity is caused by the interplay between the shape of the coefficient constraint set and that of the contours of the objective

function. The examples of the preceding section fall into two main categories distinguished by the convexity of their constraint sets.

4.1 Non-Convex Constraint Set

BSR has a non-convex constraint set for $k < p$, and Subsection 3.3 gives some intuition for why the DF can be much greater than the model dimension. This intuition can be generalized to any non-convex constraint set, as follows. Place the true mean at a point with non-unique projection onto the constraint set, see Figure 6 for a generic sketch. Such a point must exist by the Motzkin-Bunt Theorem (Bunt, 1934; Motzkin, 1935; Kritikos, 1938), and note the constraint set for $\hat{\boldsymbol{y}} = \boldsymbol{X}\hat{\boldsymbol{\beta}}$ is just an affine transformation of the constraint set for $\hat{\boldsymbol{\beta}}$, and thus a non-convex $\hat{\boldsymbol{\beta}}$ constraint is equivalent to a non-convex $\hat{\boldsymbol{y}}$ constraint. Then the fit depends sensitively on the noise process, even when the noise is very small, since $\boldsymbol{y}$ is projected onto multiple well-separated sections of the constraint set. Thus as the magnitude of the noise, σ, goes to zero, the variance of $\hat{\boldsymbol{y}}$ remains roughly constant. Equation (9) then tells us that DF can be made arbitrarily large, as it will be roughly proportional to σ^{-1}. We formalize this intuition in the following theorem.

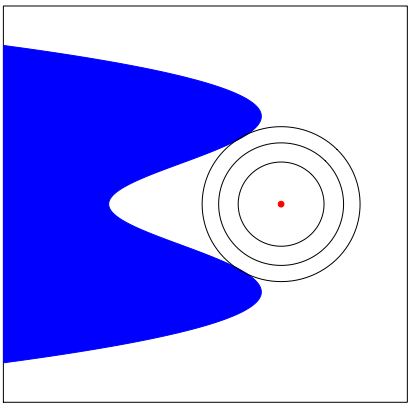

Figure 6: Sketch of geometric picture for a regression problem with $n = 2$ and non-convex constraint sets. The blue area is the constraint set, the red point the true mean vector, and the black circles the contours of the least squares objective function. Note how the spherical contour spans the gap of the divot where it meets the constraint set.

Theorem 2. *For a fitting technique FIT that minimizes squared error subject*

to a non-convex constraint $\boldsymbol{\mu} \in \mathcal{M}$, consider the model,

$$\boldsymbol{y} = \boldsymbol{\mu} + \sigma\boldsymbol{\varepsilon}, \quad \boldsymbol{\varepsilon} \overset{\text{i.i.d.}}{\sim} F,$$

where F is a mean-zero distribution with finite variance supported on an open neighborhood of 0. Then there exists some $\boldsymbol{\mu}^*$ such that $DF(\boldsymbol{\mu}^*, \sigma, FIT_k) \to \infty$ as $\sigma \to 0$.

Proof. We leave the proof for the appendix. □

4.2 Convex Constraint Set

Projection onto a convex constraint set is a contraction (Bertero and Boccacci, 1998), which limits the DF of least-squares fitting with convex constraints. If convex set $\mathcal{M}$ lies in a p-dimensional affine space $\mathcal{V}$ (e.g. if we fit a linear model with a convex constraint on the coefficients), then the DF can never exceed p. To see this, first note that

$$\hat{\boldsymbol{y}} = \mathcal{P}_{\mathcal{M}}(\boldsymbol{y}) = \mathcal{P}_{\mathcal{M}}(\mathcal{P}_{\mathcal{V}}(\boldsymbol{y})), \tag{12}$$

and consider the covariance formula for DF in Equation (9). For the DF to go above p, the variance of $\hat{\boldsymbol{y}}$ must be greater than that of $\mathcal{P}_{\mathcal{V}}\boldsymbol{y}$ in at least one direction, which cannot happen since $\hat{\boldsymbol{y}}$ is a contraction of $\boldsymbol{y}$.

However, even with convex sets, the DF can still be non-monotone. The Lasso example is instructive, as we can draw it in two dimensions, see Figure 7. Since the noise is very low, the first intersection of the contours of the objective function around the true $\boldsymbol{\beta}$ (pictured) with the constraint set give the nearly deterministic value of the estimated coefficient vector. Since the green and orange cases (ℓ_1 norms of 0.9 and 3.3, respectively) have the intersection at a corner of the constraint set, the DF for those problems will be around zero. The blue case, however, projects to a face of the constraint set, a one-dimensional set, corresponding to a DF of around one. These DF correspondences can be figured from the formulae in Kato (2009). Therefore it is the deletion of the second variable at around $||\hat{\boldsymbol{\beta}}||_1 = 3.3$ that accounts for the non-monotonicity in this case.

Note that the Lasso example still relies on some irregularity in the constraint contours, namely the sharp corners. Alternatively, we could consider ridge regression (possibly with unequal penalties on the coefficient indices), whose constraint contours are elliptical and thus quite regular in some sense. Indeed, the solution to penalized ridge regression has a closed form, linear solution, so its DF is easily computable using Equation (10) and is monotone with respect to inclusion (this result is also proved formally in Kaufman and Rosset (2013)). Surprisingly however, DF monotonicity can still be lost in the constrained form of ridge regression, as shown by an ingenious example in Kaufman and Rosset (2013). Those authors take advantage of the fact that nested ellipses (unlike nested spheres or ℓ_1 balls) are not parallel curves, highlighting the importance of irregular contour shapes to produce the non-monotonicity in DF.

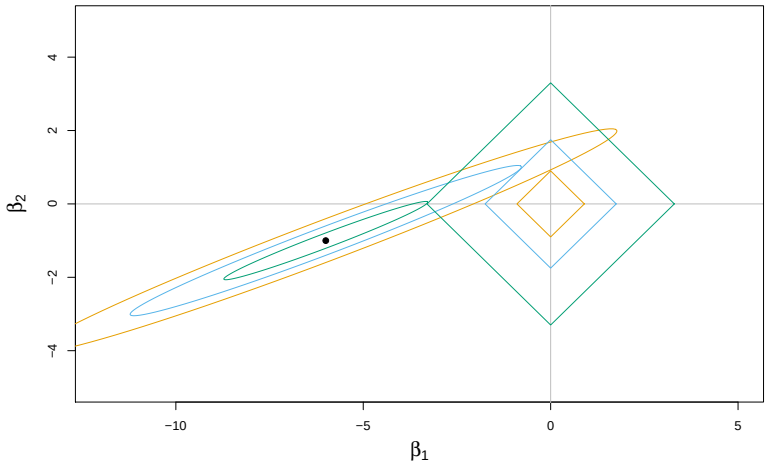

Figure 7: Picture for the Lasso example in Subsection 3.2. The black point is the true coefficient vector. Note that both the orange contour and the green contour meet the constraint set at the corners of their respective ℓ_1 balls.

5 Discussion

The common intuition that "effective" or "equivalent" degrees of freedom serves a consistent and interpretable measure of model complexity merits some degree of skepticism. In particular, we show that for many widely-used fitting tunable techniques, the DF can be non-monotone with respect to model nesting, and can exceed the dimension of the model space by an arbitrarily large amount. This phenomenon is not restricted to pathological examples or edge cases, but can arise in run-of-the-mill datasets as well. Finally, we presented a geometric interpretation of this phenomenon in terms of irregularities in the model constraints and/or objective function contours.

In light of the above, we find the term "(effective) degrees of freedom" to be somewhat misleading, as it is suggestive of a quantity corresponding to model "size". It is also misleading to consider DF as a measure of overfitting, or how flexibly the model is conforming to the data, since a model always fits at least as strongly as a strict submodel to a given dataset. By definition, the DF of Efron (1983) does measure optimism of in-sample error as an estimate of out-of-sample error, but we should not be too quick to assume that our intuition about linear models carries over exactly.

References

Bertero, M. and Boccacci, P. (1998). Introduction to Inverse Problems in Imaging. In *Physics*, volume First, chapter Appendix F. Taylor & Francis.

Buja, A., Hastie, T., and Tibshirani, R. (1989). Linear smoothers and additive models. *The Annals of Statistics*, 17(2):453–510.

Bunt, L. N. H. (1934). *Bijdrage tot de theorie der convexe puntverzamelingen.* PhD thesis, Rijks-Universiteit, Groningen.

Efron, B. (1983). Estimating the error rate of a prediction rule: improvement on cross-validation. *Journal of the American Statistical Association*, 78(382):316–331.

Efron, B. (1986). How Biased Is the Apparent Error Rate of a Prediction Rule? *Journal of the American Statistical Association*, 81(394):461–470.

Hoerl, A. E. (1962). Application of Ridge Analysis to Regression Problems. *Chemical Engineering Progress*, 58:54–59.

Kato, K. (2009). On the degrees of freedom in shrinkage estimation. *Journal of Multivariate Analysis*, 100(7):1338–1352.

Kaufman, S. and Rosset, S. (2013). When Does More Regularization Imply Fewer Degrees of Freedom? Sufficient Conditions and Counter Examples from Lasso and Ridge Regression. *arXiv preprint arXiv:1311.2791*.

Kritikos, M. N. (1938). Sur quelques propriétés des ensembles convexes. *Bulletin Mathématique de la Société Roumaine des Sciences*, 40:87–92.

Mallows, C. L. (1973). Some Comments on CP. *Technometrics*, 15(4):661–675.

Motzkin, T. (1935). Sur quelques propriétés caractéristiques des ensembles convexes. *Atti Accad. Naz. Lincei Rend. Cl. Sci. Fis. Mat. Natur*, 21:562–567.

Tibshirani, R. (1996). Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B*, 58(1):267–288.

Tibshirani, R. J. and Taylor, J. (2012). Degrees of freedom in lasso problems. *The Annals of Statistics*, 40(2):1198–1232.

Zou, H., Hastie, T., and Tibshirani, R. (2007). On the degrees of freedom of the lasso. *The Annals of Statistics*, 35(5):2173–2192.

6 Appendix A

Proof of Theorem 2. The proof relies heavily on the fact that for every non-convex set $\mathcal{M}$ in Euclidean space there is at least one point whose projection onto $\mathcal{M}$ is not unique (e.g. the red dot in Figure 6). This fact was proved independently in Bunt (1934), Motzkin (1935), and Kritikos (1938). A schematic for this proof in two dimensions is provided in Figure 8. Let $\boldsymbol{\mu}^*$ be a point

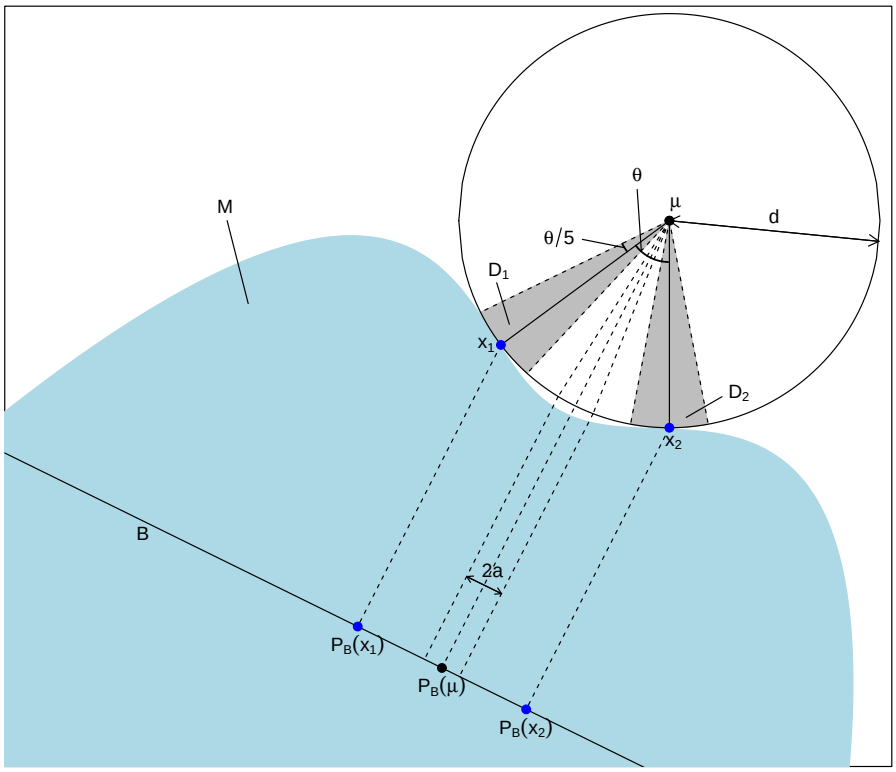

Figure 8: Schematic for the proof of Theorem 2, in two dimensions.

with non-unique projection onto the non-convex set $\mathcal{M}$ and let $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$ be two distinct projections of $\boldsymbol{\mu}$ onto $\mathcal{M}$. Let $d = ||\boldsymbol{\mu} - \boldsymbol{x}_1||_2 = ||\boldsymbol{\mu} - \boldsymbol{x}_2||_2$ be the Euclidean distance between $\boldsymbol{\mu}$ and $\mathcal{M}$, and $\theta = \cos^{-1}(\frac{(\boldsymbol{x}_1-\boldsymbol{\mu})\cdot(\boldsymbol{x}_2-\boldsymbol{\mu})}{|\boldsymbol{x}_1-\boldsymbol{\mu}||\boldsymbol{x}_2-\boldsymbol{\mu}|})$ be the angle between $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, taken as vectors from $\boldsymbol{\mu}$. Define the set $\mathcal{D}_1 = \{\boldsymbol{v} \in \mathbb{R}^n | \cos^{-1}(\frac{(\boldsymbol{x}_1-\boldsymbol{\mu})\cdot(\boldsymbol{v}-\boldsymbol{\mu})}{|\boldsymbol{x}_1-\boldsymbol{\mu}||\boldsymbol{v}-\boldsymbol{\mu}|}) < \frac{\theta}{5}, ||\boldsymbol{v}-\boldsymbol{\mu}||_2 < d\}$, and $\mathcal{D}_2$ analogously for $\boldsymbol{x}_2$. Let $\mathcal{B}$ be a one-dimensional affine subspace that is both parallel to the line connecting $\boldsymbol{x}_1$ and $\boldsymbol{x}_2$, and contained in the hyperplane defined by $\boldsymbol{\mu}$, $\boldsymbol{x}_1$, and $\boldsymbol{x}_2$. Denoting the projection operator onto $\mathcal{B}$ by $P_{\mathcal{B}}$, let $z = ||P_{\mathcal{B}}\boldsymbol{y} - P_{\mathcal{B}}\boldsymbol{\mu}||_2/\sigma$,

and $\tilde{y} = ||P_{\mathcal{B}}\hat{\boldsymbol{y}} - P_{\mathcal{B}}\boldsymbol{\mu}||_2$. Let $a = d\cos(\frac{\pi - \theta/5}{2})$. We now have,

$$\begin{aligned}
\operatorname{tr}(\operatorname{Cov}(\boldsymbol{y}, \hat{\boldsymbol{y}})) &\geq \operatorname{Cov}(P_{\mathcal{B}}\boldsymbol{y}, P_{\mathcal{B}}\hat{\boldsymbol{y}}) \\
&= \mathbb{E}[\sigma z \tilde{y}] \\
&= \mathbb{E}[\sigma z \tilde{y} 1_{y \in \mathcal{D}_1 \cup \mathcal{D}_2}] + \mathbb{E}[\sigma z \tilde{y} 1_{y \notin \mathcal{D}_1 \cup \mathcal{D}_2}] \\
&\geq \sigma \mathbb{E}[z \tilde{y} 1_{y \in \mathcal{D}_1}] + \sigma \mathbb{E}[z \tilde{y} 1_{y \in \mathcal{D}_2}] \\
&\geq a\sigma\big(\mathbb{E}[z 1_{y \in \mathcal{D}_2}] - \mathbb{E}[z 1_{y \in \mathcal{D}_1}]\big), \\
&= 2a\sigma \mathbb{E}[z 1_{y \in \mathcal{D}_2}],
\end{aligned}$$

where the first inequality follows from the translation and rotation invariance of the trace of a covariance matrix, and from the positivity of the diagonal entries of the covariance matrix for the case of projection fitting methods. The second inequality follows again from the positivity of the DF of projection fitting methods (applied to the same model with a noise process that has support on $\mathcal{D}_1$ and $\mathcal{D}_2$ removed), ensuring the second term in the third line is non-negative. The third inequality follows from considering the projections of $\mathcal{D}_1$ and $\mathcal{D}_2$ onto $\mathcal{M}$ and then onto $\mathcal{B}$, and noting that the two (double) projections must be separated by at least a distance of $2a$.

Defining $\mathcal{F}_1 = \{\boldsymbol{v} \in \mathbb{R}^n | \cos^{-1}(\frac{(\boldsymbol{x}_1 - \boldsymbol{\mu}) \cdot (\boldsymbol{v} - \boldsymbol{\mu})}{|\boldsymbol{x}_1 - \boldsymbol{\mu}||\boldsymbol{v} - \boldsymbol{\mu}|}) < \frac{\theta}{5}\}$ and $\mathcal{F}_2$ analogously for $\boldsymbol{x}_2$, note that $\mathbb{P}(\boldsymbol{y} \in \mathcal{F}_1 \setminus \mathcal{D}_1) = \mathbb{P}(\boldsymbol{y} \in \mathcal{F}_2 \setminus \mathcal{D}_2) \to 0$ as $\sigma \to 0$. Thus,

$$\operatorname{tr}(\operatorname{Cov}(\boldsymbol{y}, \hat{\boldsymbol{y}})) \geq 2a\sigma\big(\mathbb{E}[z 1_{y \in \mathcal{F}_2}] + o(\sigma)\big).$$

Neither z nor the event $y \in \mathcal{F}_2$ depend on σ, so define the constant $b = 2a\mathbb{E}[z 1_{y \in \mathcal{F}_2}] > 0$ which is independent of σ. Thus we have shown that,

$$\begin{aligned}
\operatorname{DF}(\boldsymbol{\mu}^*, \sigma, \operatorname{FIT}_k) &= \frac{1}{\sigma^2} \operatorname{tr}(\operatorname{Cov}(\boldsymbol{y}, \hat{\boldsymbol{y}})) \\
&\geq \frac{b + o(\sigma)}{\sigma} \\
&\to \infty
\end{aligned}$$

as $\sigma \to 0$. □

7 Appendix B

In Subsections 3.1 and 3.3, DF is estimated by computing an ubiased estimator of DF for each simulated noise realization. This unbiased estimator for DF can be obtained from Equation (9) by exploiting the linearity of the expectation and trace operators,

$$\begin{aligned}
\operatorname{DF}(\boldsymbol{\beta}, \boldsymbol{X}, \operatorname{FIT}_k) &= \frac{1}{\sigma^2} \operatorname{tr}(\operatorname{Cov}(\boldsymbol{y}, \hat{\boldsymbol{y}}^{(FIT_k)})) \\
&= \frac{1}{\sigma^2} \mathbb{E}\big[(\boldsymbol{y} - \boldsymbol{\mu})^T (\hat{\boldsymbol{y}}^{(FIT_k)} - \mathbb{E}[\hat{\boldsymbol{y}}^{(FIT_k)}])\big] \qquad (13) \\
&= \frac{1}{\sigma^2} \mathbb{E}\big[\boldsymbol{\varepsilon}^T \hat{\boldsymbol{y}}^{(FIT_k)}\big],
\end{aligned}$$

where the last inequality follows because $\mathbb{E}[\boldsymbol{\varepsilon}] = \mathbf{0}$. Thus the true DF is estimated by $\boldsymbol{\varepsilon}^T \hat{\boldsymbol{y}}^{(FIT_k)}$ averaged over many simulations. Since this estimate of DF is an average of *i.i.d.* random variables, its standard deviation can be estimated by the emipirical standard deviation of the $\boldsymbol{\varepsilon}^T \hat{\boldsymbol{y}}^{(FIT_k)}$ divided by the square root of the number of simulations. For Subsection 3.2, the low model noise caused the DF estimate from Equation (13) to be very noisy. Instead, the 1000 simulations were split into 10 blocks, and for each block Equation (9) was used to estimate the DF, with uncertainty quantified by the standard error over the 10 blocks.

The following is the code for the BSR simulation in Subsection 3.1. Note that although the seed is preset, this is only done to generate a suitable design matrix, not to cherry-pick the simulated noise processes. The simulation results remain the same if the seed is reset after assigning the value to `x`.

```
library(leaps)
library(gplots)
n = 50; p = 15; B = 5000
set.seed(1113)
x = matrix(rnorm(n * p), n, p)
mu = drop(scale((x %*% rep(1, 15))))
snr = 7
mu = mu * snr
dfmat = matrix(0, B, p)
for(j in 1:B){
  y = rnorm(n) + mu
  #temp = regsubsets(x, y, nbest = 1, nvmax = p)
        temp = regsubsets(x, y, nbest = 1, nvmax = p, intercept = FALSE)
  for(i in 1:p){
    jcoef = coef(temp, id = i)
    #xnames = names(jcoef)[-1]
                xnames = names(jcoef)
    which = match(xnames, letters[1:p])
    #yhat = cbind(1, x[ , which]) %*% jcoef
                if(i == 1){
                    yhat = matrix(x[ , which], n, 1) %*% jcoef
                } else{
                    yhat = x[ , which] %*% jcoef
                }
    dfmat[j, i] = sum((y - mu) * yhat)
  }
}
#df = apply(dfmat, 2, mean) - 1
df = apply(dfmat, 2, mean)
error = sqrt(apply(dfmat, 2, var) / B)
save(df, error, p, file = "BSR.rdata")
plotCI(1:p, df, 2 * error, ylim = c(0, 35), xlim = c(0, p), gap = 0.4, xlab =
     "Subset Size", ylab = "Degrees of Freedom", cex.lab = 1.5, type = "o")
abline(h = 15, lty = 2)
lines(c(0, 15), c(0, 15), lty = 2)
title("Best Subset Regression", cex.main = 2)
```

The code for the FSR simulation in Subsection 3.1 is almost identical.

```
library(leaps)
library(gplots)
n = 50; p = 15; B = 5000
set.seed(1113)
x = matrix(rnorm(n * p), n, p)
mu = drop(scale((x %*% rep(1, 15))))
snr = 7
mu = mu * snr
dfmat = matrix(0, B, p)
for(j in 1:B){
```

```
  y = rnorm(n) + mu
  #temp = regsubsets(x, y, nbest = 1, nvmax = p, method = "forward")
        temp = regsubsets(x, y, nbest = 1, nvmax = p, method = "forward",
            intercept = FALSE)
  for(i in 1:p){
    jcoef = coef(temp, id = i)
    #xnames = names(jcoef)[-1]
                xnames = names(jcoef)
    which = match(xnames, letters[1:p])
    #yhat = cbind(1, x[ , which]) %*% jcoef
                if(i == 1){
                    yhat = matrix(x[ , which], n, 1) %*% jcoef
                } else{
                    yhat = x[ , which] %*% jcoef
                }
                dfmat[j, i] = sum((y - mu) * yhat)
  }
}
#df = apply(dfmat, 2, mean) - 1
df = apply(dfmat, 2, mean)
error = sqrt(apply(dfmat, 2, var) / B)
save(df, error, p, file = "FSR.rdata")
plotCI(1:p, df, 2 * error, ylim = c(0, 35), xlim = c(0, p), gap = 0.4, xlab =
      "Subset Size", ylab = "Degrees of Freedom", cex.lab = 1.5, type = "o")
abline(h = 15, lty = 2)
lines(c(0, 15), c(0, 15), lty = 2)
title("Forward Stepwise Regression", cex.main = 2)
```

The code for Subsection 3.2 is a little more involved, and relies on the glmnet package in R.

```
# B = 1000: ~7.5 min
library(glmnet)
library(gplots)

X = matrix(c(0, 2, 1, -5), 2)
XtX.inv = solve(t(X) %*% X)
true.beta = c(-6, -1)
s = 0.03

B = 1000
y.true = X %*% true.beta
sol.path = glmnet(X, y.true, standardize = FALSE, lambda = exp(seq(3, -5,
     -0.001)), intercept = FALSE)
grid.c = c(seq(0, 3, 0.2), seq(3, 4, 0.05), seq(4, 6, 0.2))
beta.hat = matrix(NA, 2, length(grid.c))
yhat.lasso.path = array(NA, dim = c(length(grid.c), B, 2))
y = matrix(NA, 2, B)
for(i in 1:B) {
    if(i %% 20 == 0) print(i)
    y[ , i] = X %*% true.beta + s * rnorm(2)

    sol.path = glmnet(X, y[ , i], standardize = FALSE, lambda = exp(seq(3,
         -5, -0.001)), intercept = FALSE)
    beta.hat[1, ] = approx(colSums(abs(sol.path$beta)), sol.path$beta[1, ],
         xout = grid.c)$y
    beta.hat[2, ] = approx(colSums(abs(sol.path$beta)), sol.path$beta[2, ],
         xout = grid.c)$y
    yhat.lasso.path[ ,i , ] = t(X %*% beta.hat)
}
pdf("df_Lasso.pdf", height = 6.5, width = 10)
par(mfrow = c(2, 1), mar = c(4.1, 4.1, 3.1, 1.2))
y.true = X %*% true.beta
sol.path = glmnet(X, y.true, standardize = FALSE, lambda = exp(seq(3, -5,
     -0.001)), intercept = FALSE)
plot(colSums(abs(sol.path$beta)), sol.path$beta[1, ], type = "l", xlim = c(0,
      6), ylim = c(min(sol.path$beta[1, ]), max(sol.path$beta[2, ])), main =
     "", xlab = "L1 Constraint", ylab = "Coefficients", cex.lab = 1.5)
```

```
points(colSums(abs(sol.path$beta)), sol.path$beta[2, ], type = "l", col = "
    red")
grid.c = c(seq(0, 3, 0.2), seq(3, 4, 0.05), seq(4, 6, 0.2))
zeros = (1:8001)[sol.path$beta[2,]==0 & colSums(abs(sol.path$beta)) > 0]
abline(h = 0, lty = 3)
abline(v = colSums(abs(sol.path$beta))[min(zeros)], lty = 3)
abline(v = colSums(abs(sol.path$beta))[max(zeros)], lty = 3)

m = 10
df = sapply(1:length(grid.c), function(i){
  (1/s^2) * mean(sapply(1:m, function(j){
    ind = ((j-1)*B/m+1):(j*B/m)
    cov(yhat.lasso.path[i, ind, 1], y[1, ind]) +
      cov(yhat.lasso.path[i, ind, 2], y[2, ind])}))})
std = sapply(1:length(grid.c), function(i){
  (1/(sqrt(m)*s^2)) * sd(sapply(1:m, function(j){
    ind = ((j-1)*B/m+1):(j*B/m)
    cov(yhat.lasso.path[i, ind, 1], y[1, ind]) +
      cov(yhat.lasso.path[i, ind, 2], y[2, ind])}))})
plotCI(grid.c, df, 2*std, type = 'l', gap = 0.1, ylab = "Degrees of Freedom",
     xlab = "L1 Constraint", cex.lab = 1.5)
abline(v = colSums(abs(sol.path$beta))[min(zeros)], lty = 3)
abline(v = colSums(abs(sol.path$beta))[max(zeros)], lty = 3)
dev.off()
save(sol.path, zeros, grid.c, df, std, file = "Lasso.rdata")
```

Finally, the code for Subsection 3.3 is different but short. We can imagine taking $A \to \infty$.

```
# A is some big number
A = 10000
X = A * data.frame(diag(2))
B = 1E5
y = A + matrix(rnorm(2 * B), ncol = 2)

# best univariate (no-intercept) model just picks the larger y_i
yhat.bs = t(apply(y, MARGIN = 1, FUN = function(yy) yy * (yy > mean(yy))))

dfvec = drop(((y - A) * yhat.bs) %*% c(1, 1))
df = mean(dfvec)
error = sqrt(var(dfvec) / B)
c(df, error)
```