Inference: Subsampling and resampling approaches

Patrick Breheny

April 13

Introduction

- Today's lecture will focus on using subsampling, resampling, and sample splitting as ways to carry out inference for high-dimensional models
- These methods tend to be somewhat computationally intensive, as they can involve fitting a high-dimensional model hundreds or thousands of times
- This is not necessarily prohibitive from the standpoint of analyzing a single data set, but it does present a barrier to comprehensive simulation studies

Example data for today

To illustrate the methods, I'll apply them to a simulated data set with the same basic construction as those we looked at last week ($n = 100$, $p = 60$, $\sigma^2 = 1$):

- Six variables with $\beta_j \neq 0$ (category "A"):
  - Two variables with $\beta_j = \pm 1$:
  - Four variables with $\beta_j = \pm 0.5$:
- Each of the six variables is correlated ($\rho = 0.5$) with two other variables (i.e., 12 variables fall into this category) for which $\beta_j = 0$ ("B")
- The remaining 42 variables are pure noise, $\beta_j = 0$ and independent of all other variables ("C")

Sample splitting: Idea

- We begin with the simplest idea: sample splitting
- We have already seen the basic idea of sample splitting when we discussed the "refitted cross-validation" approach to estimating σ^2
- The approach involves two steps:
  (1) Take half of the data and fit a penalized regression model (e.g., the lasso); typically this involves cross-validation as well for the purposes of selecting λ
  (2) Use the remaining half to fit an ordinary least squares model using only the variables that were selected in step (1)

Sample splitting: Example (step 1)

- Let's split the example data set into two halves, D_1 and D_2, each with $n = 50$ observations
- Fitting a lasso model to D_1 and using cross-validation to select λ, we select 29 variables:
  - 5 from category A
  - 5 from category B
  - 19 from category C

Sample splitting: Example (step 2)

- Fitting an ordinary linear regression model to the selected variables ($n = 50$, $p = 29$), only two coefficients (the two with $\beta_j = 1$) are significant in the $p < 0.05$ sense
- If we relax that to $p < 0.1$, an additional variable with $\beta_j = 0.5$ is found to be significant, as is a variable from category "B" (as you might expect, the variable it was correlated with was the one that was not selected by the original lasso)
- We can obtain confidence intervals as well, although note that we only obtain confidence intervals for coefficients selected in step (1)

Sample splitting: Advantages and disadvantages

- The main advantage of the sample splitting approach is that it is clearly valid: all inference is derived from classical linear model theory
- One minor obstacle, as we have seen, is that one can have increased type I errors if we fail to select all of the important variables at stage (1)
- The main disadvantages are:
  - Lack of power due to splitting the sample size in half
  - Results can vary considerably depending on the split chosen

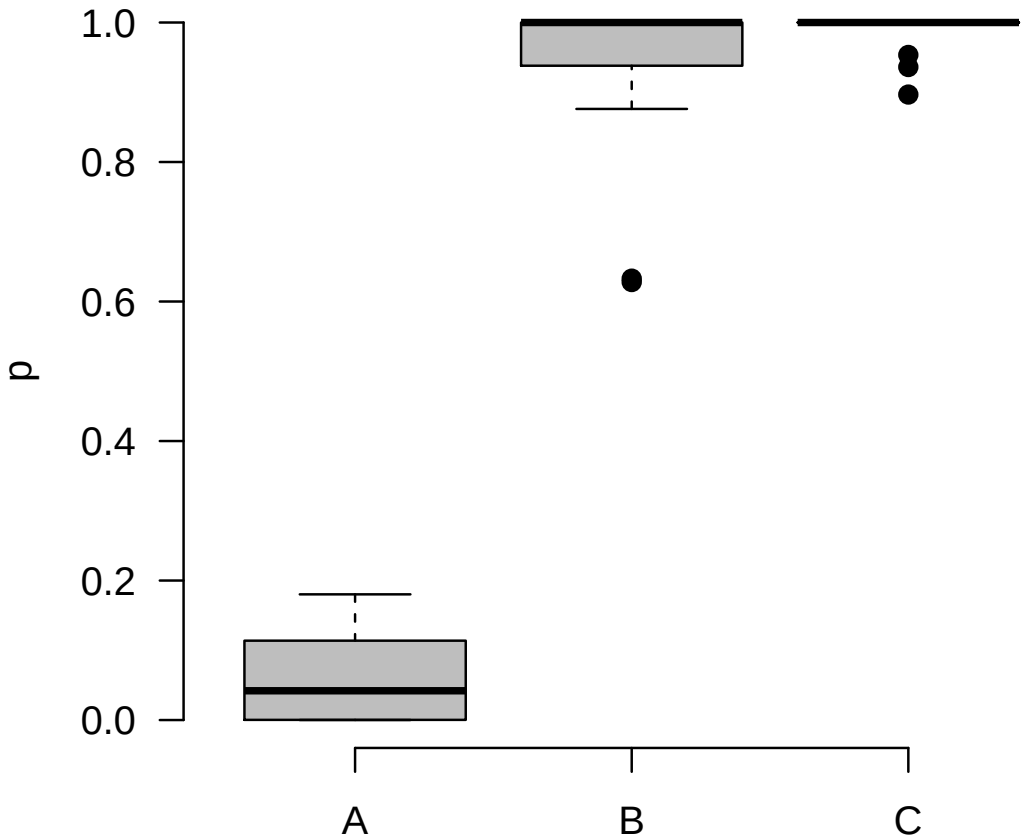
Multiple splits

- An obvious remedy for the second disadvantage is to apply the sample splitting procedure many times and average over the splits
- To some extent, this will also help with the problem of failing to select important variables in stage (1)
- One major challenge with this approach, however, is how exactly we average over results in which a covariate was not included in the model

Averaging over unselected variables

- One conservative remedy is to simply assign $p_j = 1$ whenever $j \notin \mathcal{S}$, the set of selected variables from stage 1
- With this substitution in place, we will have, for each variable, a vector of p-values $p_j^{(1)}, \ldots, p_j^{(B)}$, where B is the number of random splits, which we can aggregate in a variety of ways
- For the results that follow, I used the median

Multiple split approach applied to example data



Four variables from A have $p < 0.05$

Remarks

- Certainly, the results are much more stable if we average across sample splits
- The other downside, however, (loss of power from splitting the sample in two) cannot be avoided
- It is possible to extend this idea to obtain confidence intervals as well by inverting the hypothesis tests, although the implementation gets somewhat complicated

TCGA data

- To get a feel for how conservative this approach is, let's apply it to the TCGA data ($n = 536$, $p = 17,322$)
- Using the multiple-splitting approach, only a single variable is significant with $p < 0.05$ (one other variable has $p = 0.08$; all others are above 0.1)
- This is in sharp contrast to our results from yesterday, in which we were able to identify 52 features using the false inclusion rate approach

Stability selection

- One could argue that trying to obtain a classical p-value isn't really the right goal, that what makes sense for single hypothesis testing isn't relevant to high-dimensional modeling
- Consider, then, the idea of *stability selection* (Meinshausen & Bühlmann, 2010), in which we decide that a variable is significant if it is selected in a high proportion of penalized regression models that have been applied to "perturbed" data
- The most familiar way of perturbing a data set is via resampling (i.e., bootstrapping), although the authors also considered other ideas

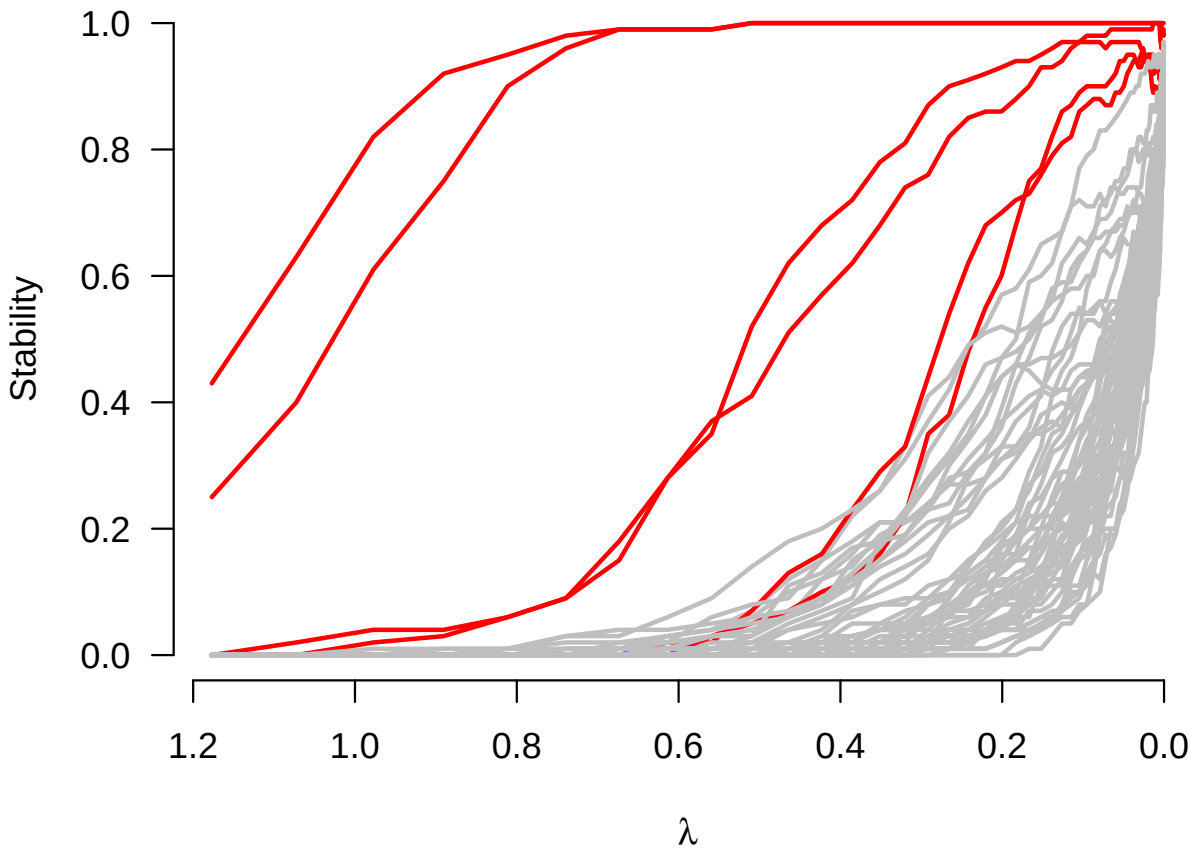
Details

- Furthermore, there are a variety of ways of carrying out bootstrapping, a point we will return to later
- For simplicity, I'll stick to what the authors chose in their original paper: randomly select $n/2$ indices from $\{1, \ldots, n\}$ without replacement (this is based on an argument from Freedman 1977 that sampling $n/2$ without replacement is fairly similar to resampling n with replacement)
- Letting π_{thr} denote a specified cutoff and $\hat{\pi}_j(\lambda)$ the fraction of times variable j is selected for a given value of λ, the set of *stable variables* is defined as

$$\{j : \hat{\pi}_j(\lambda) > \pi_{\text{thr}}\}$$

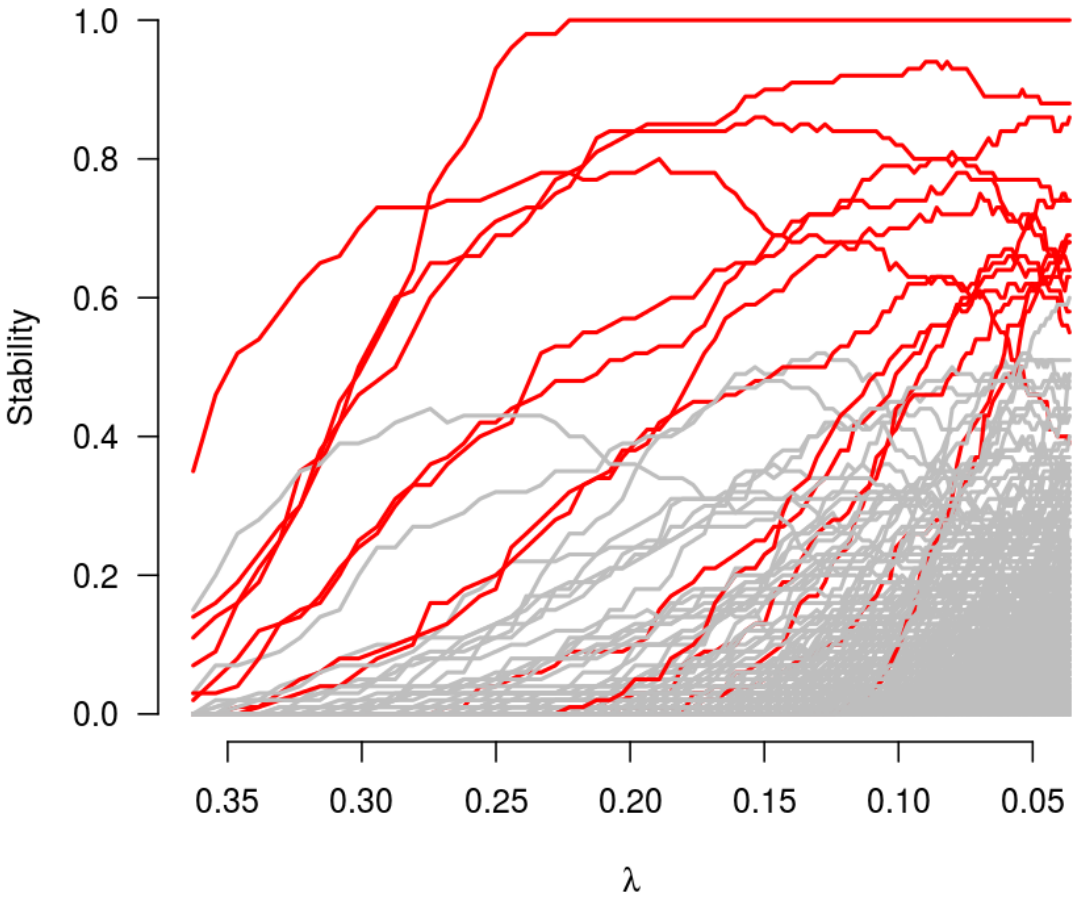
Stability selection for example data

Variables with $\beta_j \neq 0$ in red:



Stability selection for TCGA data

Variables that exceed $\pi_{\text{thr}} = 0.6$ for any λ in red:



FDR bound

- Meinshausen & Bühlmann also provide an upper bound for the expected number of false selections in the stable set (i.e., variables with $\beta_j = 0$ and $\hat{\pi}_j(\lambda) > \pi_{\text{thr}}$):

$$\frac{1}{2\pi_{\text{thr}} - 1} \frac{S(\lambda)^2}{p},$$

  where $S(\lambda)$ is the expected number of selected variables
- Note that this bound can only be applied if $\pi_{\text{thr}} > 0.5$
- In practice, however, this bound is very conservative and not particularly useful:
  - For the example data set, we identify only the two variables with $\beta_j = 1$, even if we allow an FDR of 30%
  - For the TCGA data set, no variables can be stably selected

Bootstrapping

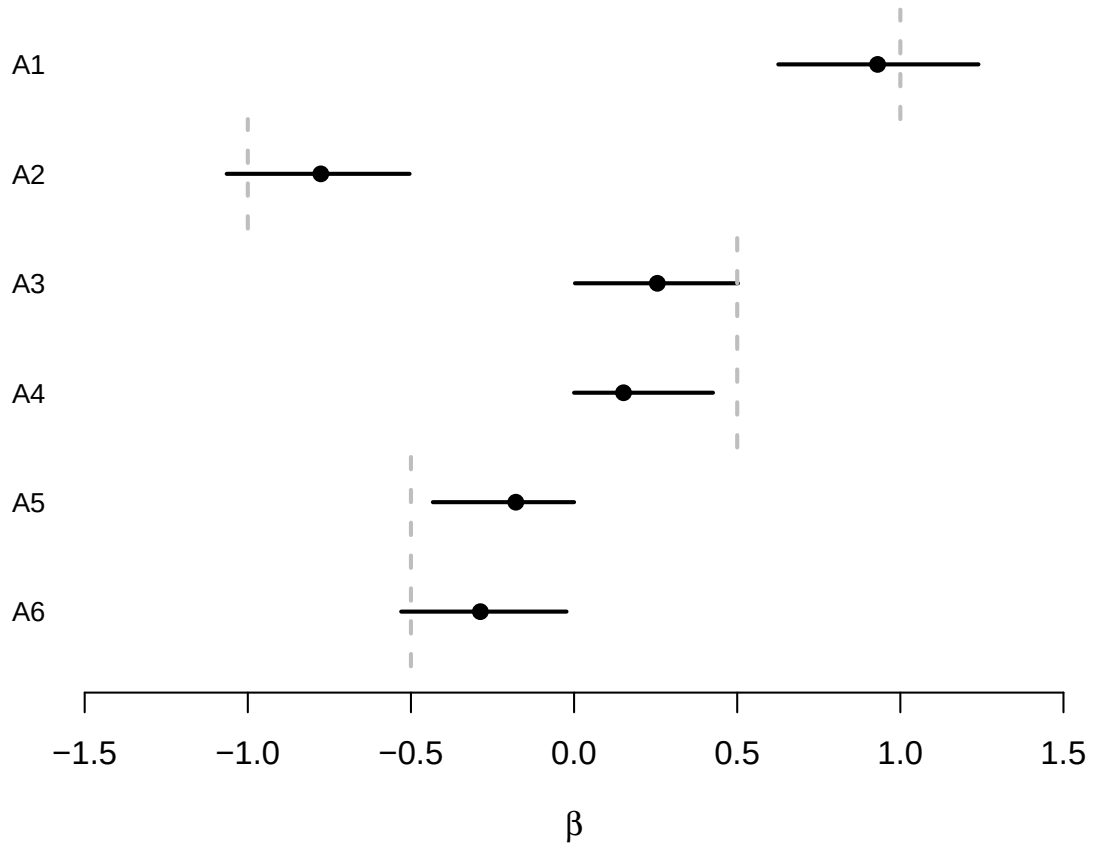
- Stability selection is essentially just bootstrapping, with a special emphasis on whether $\widehat{\beta}_j^{(b)} = 0$
- There are a variety of ways of carrying out bootstrapping for regression models; the one we have just seen, in which one selects random elements from $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, is known as the *pairs bootstrap* or *pairwise bootstrap*
- Alternatively, we may obtain estimators $\widehat{\boldsymbol{\beta}}$ and $\hat{\sigma}^2$ (e.g., from the lasso using cross-validation) and use them to bootstrap residuals parametrically:

$$\varepsilon_i^* \sim \mathrm{N}(0, \hat{\sigma}^2),$$

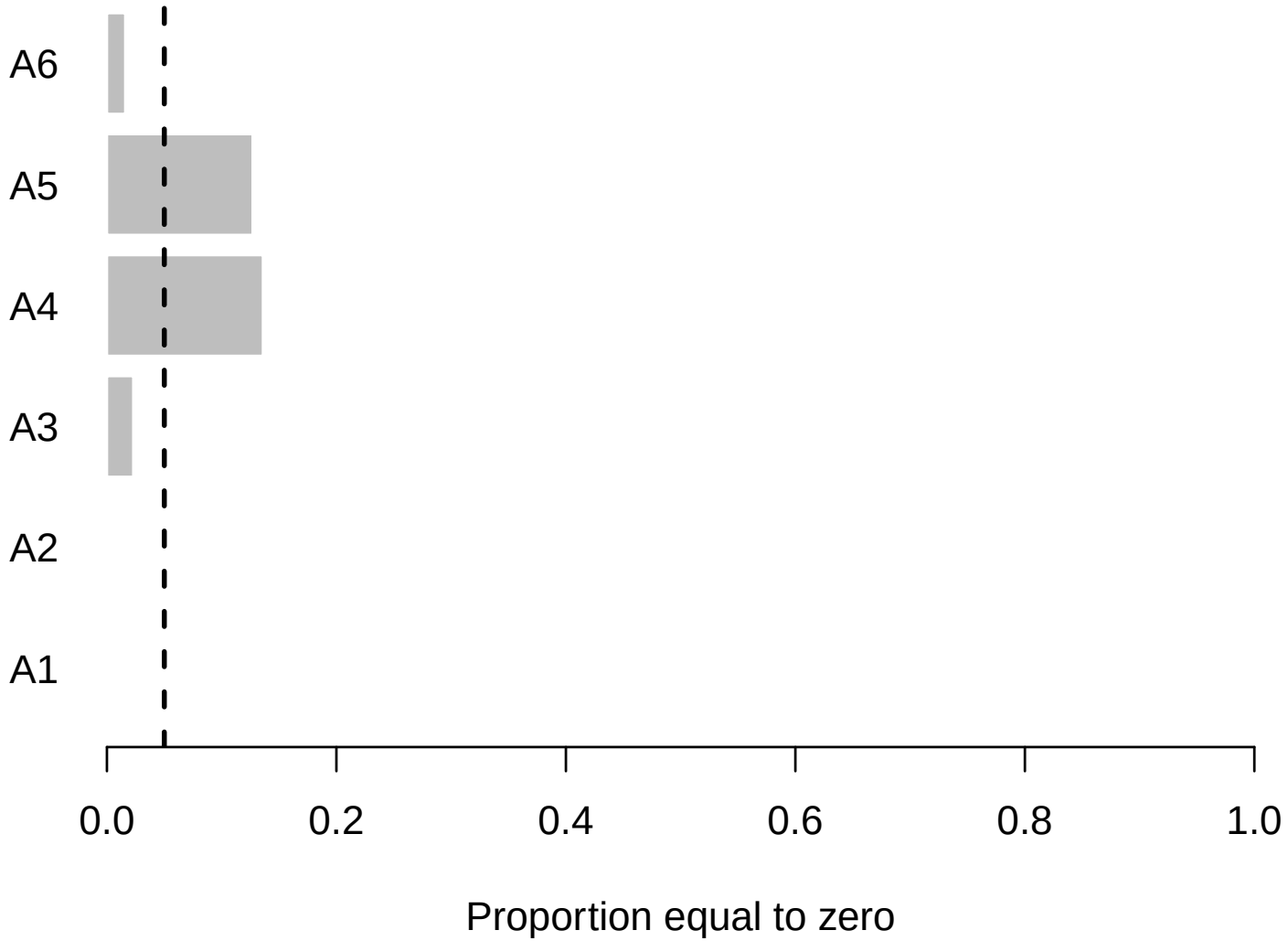
with $y_i^* = \sum_j x_{ij}\widehat{\beta}_j + \varepsilon_i^*$

Bootstrap intervals: Example data

Bootstrap percentile intervals for the six coefficients with $\beta_j \neq 0$, residual approach, λ fixed at $\hat{\lambda}_{\mathrm{CV}}$



Bootstrap and stability



Does bootstrapping work?

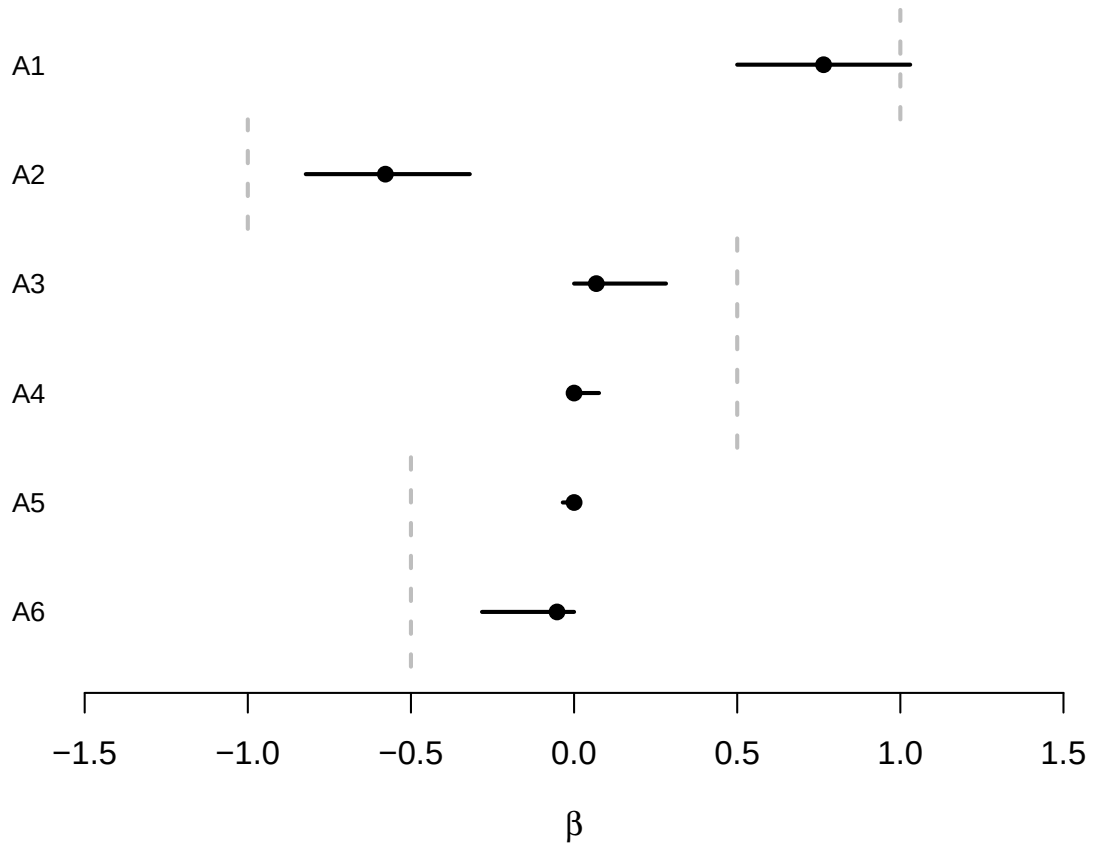
- This is interesting, but a natural question would be whether or not bootstrapping actually works in this setting
- In particular, we have theoretical results establishing that bootstrapping works for maximum likelihood; do those proofs extend to penalized likelihood settings?
- It turns out that the answer is a qualified "no"

Limitations/failures of bootstrapping

- Specifically, bootstrapping requires, at a minimum, $\sqrt{n}$-consistency
- Thus, even if it were to work with the lasso, would only work for small values of λ; i.e., $\lambda = O(1/\sqrt{n})$

Bootstrap intervals revisited

Bootstrap intervals with a larger regularization parameter, $\lambda = 0.35$:

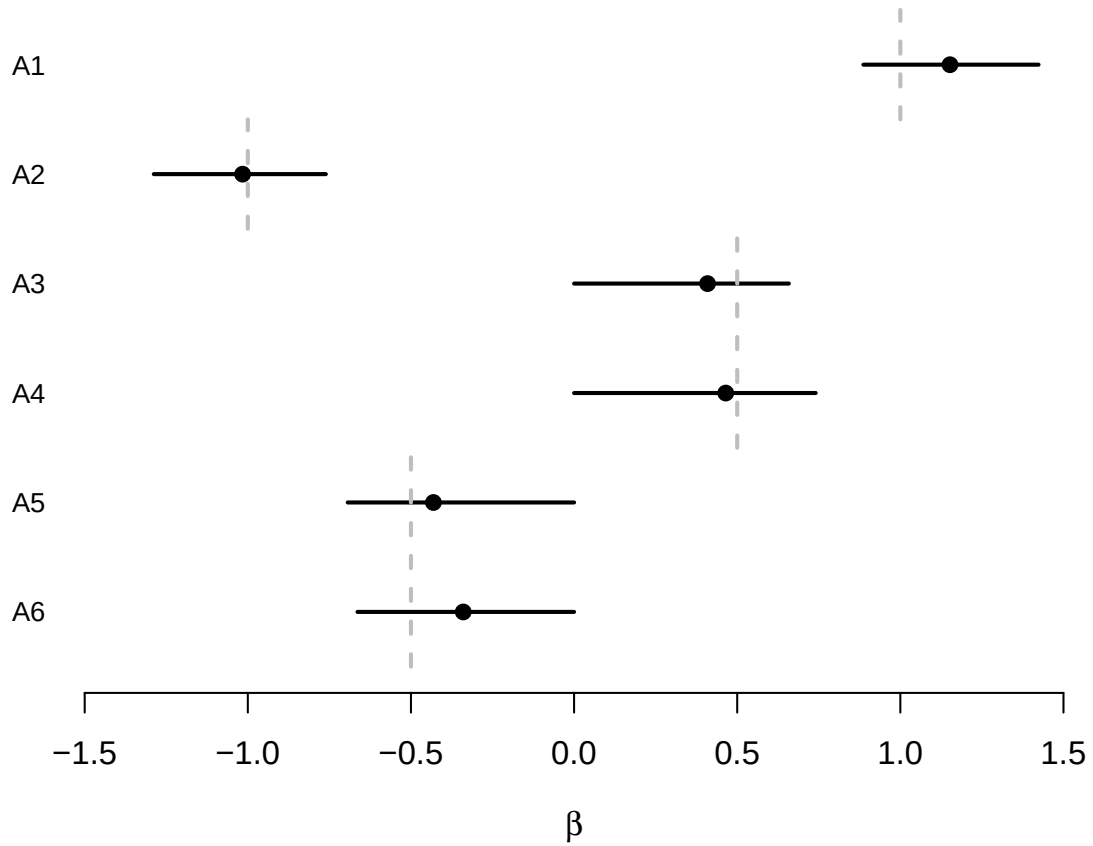


Limitations/failures of bootstrapping (cont'd)

- A subtler question is whether, even if we have $\sqrt{n}$-consistency, the bootstrap will work
- It turns out that the answer is still "no", at least for the lasso, as shown by Chatterjee and Lahiri (2010)
- However, in their follow-up paper, Chatterjee and Lahiri (2011), they show that the bootstrap does work (asymptotically) for the adaptive lasso (and by extension, other models with the oracle property, such as MCP and SCAD)
- Of course, just because it works asymptotically doesn't mean it works well in finite samples; not much work has been done in terms of rigorous simulation studies examining the accuracy of bootstrapping for MCP

Bootstrap intervals for MCP

Bootstrap percentile intervals, residual approach, λ selected by cross-validation



Bootstrap and Bayesian posterior

- Finally, it is worth noting that the distribution of bootstrap realizations $\widehat{\boldsymbol{\beta}}^*$ tends to be fairly similar to the posterior distribution of the corresponding Bayesian model in which the penalty is translated into a prior
- This raises the question, then, of whether examples like the preceding are truly failures of the bootstrap, or whether they simply reflect the incompatibility of penalization/priors and frequentist inference goals like 95% coverage