Debiasing and subsampling/resampling approaches

Patrick Breheny

April 8

Introduction

- Today's notes will discuss two unrelated approaches to inference:
  - Debiasing, in which we attempt to get around the fact that $\widehat{\beta}_j$ is biased by constructing a new statistic $\tilde{\beta}_j$ that is unbiased for β_j
  - Perturbation approaches that use subsampling, resampling, or sample splitting as ways to carry out inference for high-dimensional models
- Both of these are really categories of approaches rather than a specific approach; many ideas have been proposed that fall into each category

Debiasing

- The basic idea behind debiasing is that frequentist inference tends to work well if $\widehat{\beta}_j \dot{\sim} \mathrm{N}(\beta_j, \mathrm{SE}^2)$
- Penalized regression estimates obviously do not have this property (with the possible exception of MCP/SCAD), so debiasing approaches attempt to construct an estimate $\tilde{\beta}_j$, based on $\widehat{\boldsymbol{\beta}}$ in some way, for which approximate unbiased normality holds

Implementations

- Many authors have proposed approaches along these lines, deriving some sort of bias correction term usually along the lines of $\tilde{\beta}_j = \widehat{\beta}_j + \text{adj}$:
    - Zhang and Zhang (2014)
    - Bühlmann (2013)
    - Javanmard and Montanari (2014)
- Some of these approaches require fitting a new model (e.g., a lasso model) for each feature, and therefore are not particularly well-suited to high dimensions

Semi-penalization

- For the sake of this class, let's look at a relatively simpler way to accomplish debiasing: *semi-penalization*
- The idea here is that we can obtain a (more or less) unbiased estimate for β_j by not penalizing it; for example,

$$L(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) + \lambda \sum_{k \neq j} |\beta_k|$$

- As far as I know, this idea first appeared in Huang et al. (2013, "SPIDR"); today I'll talk about an approach proposed in Shi et al. (2019), which is very similar in concept but differs in the details

Semi-penalized LRT

- The idea here is actually very similar to the general statistical idea of a likelihood ratio test: we fit constrained and unconstrained models, and then compare their likelihoods
- Specifically, for testing $H_0 : \beta_j = 0$, we would solve for $\widehat{\boldsymbol{\beta}}_0$ that minimizes

$$L(\boldsymbol{\beta}_{-j}|\mathbf{X}_{-j}, \mathbf{y}) + \lambda \sum_{k \neq j} |\beta_k|$$

as well as $\widehat{\boldsymbol{\beta}}_a$ that minimizes

$$L(\boldsymbol{\beta}|\mathbf{X}, \mathbf{y}) + \lambda \sum_{k \neq j} |\beta_k|$$

Distribution

- It can be shown that (with a number of assumptions), the test statistic

$$2\{\ell(\widehat{\boldsymbol{\beta}}_a, \hat{\sigma}^2) - \ell(\widehat{\boldsymbol{\beta}}_0, \hat{\sigma}^2)\}$$

  follows an approximate χ^2 distribution with 1 degree of freedom, where $\ell(\boldsymbol{\beta}, \sigma^2)$ denotes the likelihood
- The error variance can be estimated using any of the methods we have discussed in class, but as in the classical LRT, is based on the unrestricted (alternative) model
- The paper discusses score and Wald tests as well, but we'll only look at the LRT

Remarks

- One of the conditions required to show convergence to the proper distribution is that $\sqrt{n}p'(\beta_j^*) \to 0$ for all $j \in \mathcal{S}$
- This is satisfied for MCP/SCAD, but not the lasso; nevertheless, it seems to me to work reasonably well for the lasso also, so I will go ahead and show those results
- This approach would also seem amenable to constructing confidence intervals, although the article doesn't discuss this
- Another thing that is not discussed in the article is whether a multiple comparison procedure is needed for the p-values, or whether the penalized regression part takes care of this automatically somehow . . . this is not entirely clear to me, so I'll just present the unadjusted p-values

Results: Example data set

	Estimate	mfdr	SPLRT
A1	0.85	$<$0.0001	$<$0.0001
A2	-0.86	$<$0.0001	$<$0.0001
A6	-0.45	$<$0.0001	$<$0.001
A4	-0.47	$<$0.0001	$<$0.001
A3	0.37	$<$0.001	$<$0.001
B9	0.22	0.04	1.00
A5	0.18	0.20	0.20
N2	0.13	0.45	0.09
N10	0.08	0.92	0.10
N17	0.00	0.96	0.38

Comments

- Results seem more or less similar for the noise variables and most of the "A" variables
- However, B9 and A5 illustrate the key difference:
  - We have convincing evidence that one of them is important according to the marginal approach, which isn't concerned about the possibility of indirect associations
  - This is a major concern for conditional approaches, however – neither variable shows up as significant in the semi-penalized LRT

High-dimensional example: TCGA

- Like several conditional approaches, the semi-penalized LRT works nicely in many low- to medium-dimensional situations, but dramatically loses power in high-dimensional data
- For example, in applying the test to our TCGA data, no genes could be identified as significant: the minimum p-value was 0.13 even without any adjustments for multiple comparisons
- In contrast, 95 features are selected via cross-validation, and 16 of those have a local mfdr under 10%

Sample splitting: Idea

- The rest of today's lecture will focus on using subsampling, resampling, and sample splitting as ways to carry out inference for high-dimensional models
- 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

Sample splitting: Idea (cont'd)

Sample splitting involves two basic 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 $(n = 50, p = 60)$ and using cross-validation to select λ, we select 10 variables:
    - 4 from category A
    - 2 from category B
    - 4 from category N

Sample splitting: Example (step 2)

- Fitting an ordinary linear regression model to the selected variables ($n = 50$, $p = 10$):
  - Three "A" features are significant in the $p < 0.05$ sense
  - One "N" feature was also significant ($p = 0.02$)
- 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
- The main disadvantages are:
  - Lack of power due to splitting the sample size in half
  - Potential increase in type I error if important variables are missed in the first stage
  - Results can vary considerably depending on the split chosen

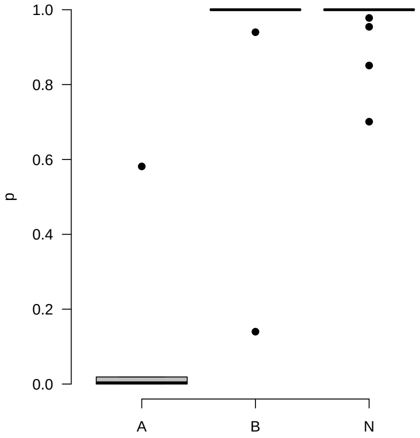
Multiple splits

- An obvious remedy for this final 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



As with the semi-penalized LRT, five "A" variables are significant

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 similar to the semi-penalized LRT, but again in sharp contrast to the marginal results

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

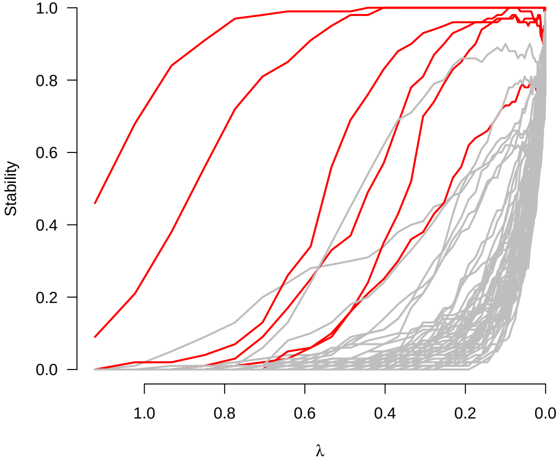
Details

- 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

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

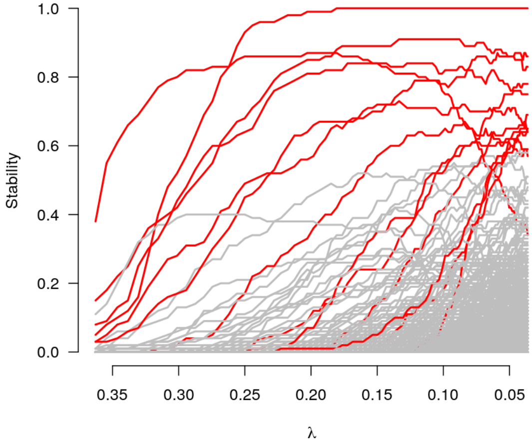
Stability selection for example data

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



Stability selection for TCGA data

13 variables 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 rather conservative:
  - For the example data set, only the two variables with $\beta_j = 1$ can be selected at an FDR of 10%; another "A" variable can be selected if we allow an FDR of 30%
  - For the TCGA data set, one variable can be stably selected

Bootstrapping

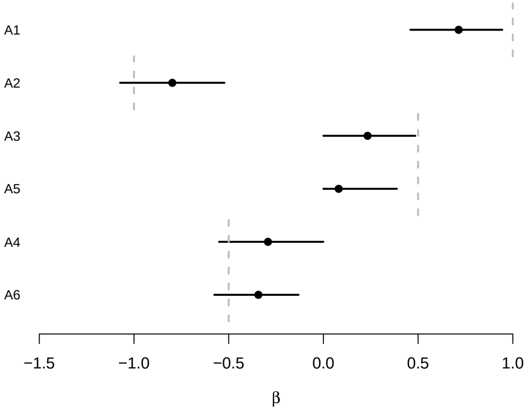
- Stability selection is essentially just bootstrapping, with a special emphasis on whether $\widehat{\beta}_j^{(b)} = 0$
- There are a variety of ways of carrying out bootstrapping for regression models; the one we have just seen, in which one selects random elements from $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, 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:

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

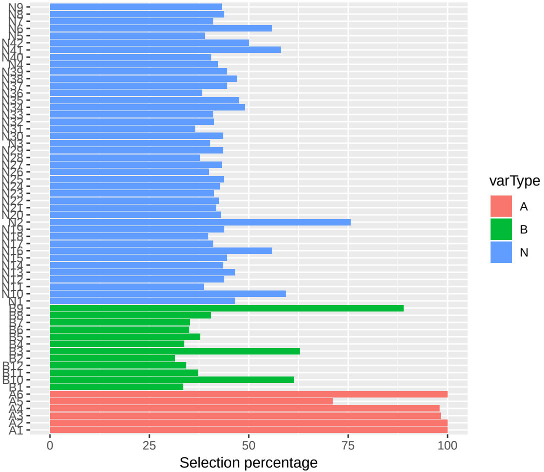
with $y_i^* = \sum_j x_{ij}\widehat{\beta}_j + \varepsilon_i^*$

Bootstrap intervals: Example data

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



Bootstrap and stability



Does bootstrapping work?

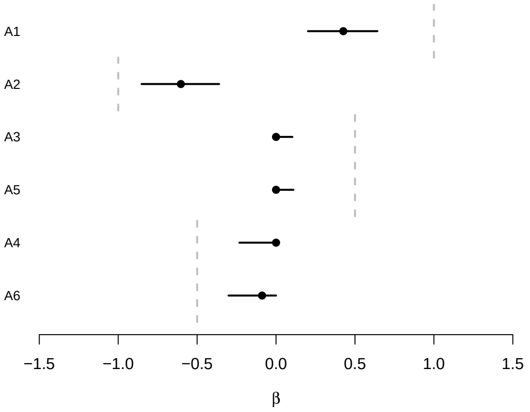
- This is interesting, but a natural question would be whether or not bootstrapping actually works in this setting
- In particular, we have theoretical results establishing that bootstrapping works for maximum likelihood; do those proofs extend to penalized likelihood settings?
- It turns out that the 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, $\lambda = 0.35$:

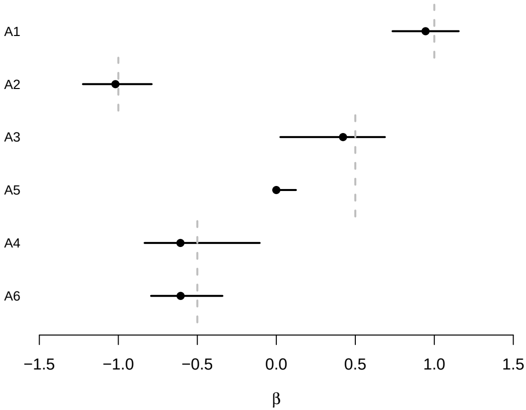


Limitations/failures of bootstrapping (cont'd)

- 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 methods with the oracle property such as adaptive lasso, 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