Multicarving

Filip Ilic & Paul Schlossmacher

2024-03-13

Questions for Christoph:

Choice of variance estimator for normalizing

When normalizing the data at the beginning, we did it with the estimator of the standard deviation, which divides by n instead of n-1, because Prof. Bühlmann did it like this in his lecture. Is this correct and does it have any consequences in the following? Maybe incompatibility with other packages, which use a different estimator? I'm guessing that it shouldn't be an issue because the whole columns are still the same up to multiplicity regardless of the method, but I'm not sure.

Regarding n_A/n_B :

On p. 3, Drysdale writes that the group A gets used for screening (i.e is the bigger group) and that

$$\hat{\beta}^{Carve} = w_A * \hat{\beta}^{Split} + w_B * \hat{\beta}^{Posi}$$

But in Lemma 3.2 on p. 4 he writes in the definition of $\hat{\beta}_j$:

$$n_B * \eta^T_{B,M_j} y_B$$

So here it seems like in fact the coefficient $\beta_j^{Split} = \eta^T_{B,M_j} y_B$ gets multiplied with the smaller set of the split, i.e. group B.

Regarding the choice of τ_M^2 in Lemma 3.2

In Lemma 3.2 Drysdale implements σ_1^2 with one τ_M^2 for both the POSI and the SPLIT part. In our implementation we chose $\tau_M^2 = \sigma^2$ with σ^2 assumed to be known and $y \sim N(X\beta^0, \sigma^2 I_n)$. However in his code of _lasso.py on row 302, Drysdale uses two different τ for POSI and SPLIT: τ_M for τ_1 (for the distribution of β^{SPLIT}), but uses some scaled version for τ_2 (for the truncated distribution of β^{POSI}). The choice of this scaling is unclear to us.

Regarding $V^{-(z)}/V^{+(z)}$:

When calculating the truncation limits $V^-(z)$ and $V^+(z)$, we tried to do it similarly to what Drysdale does in his code. Namely, we take a normalized row of the Moore Penrose Inverse of X_{M_A} together with the sign of $\hat{\beta}$ as the direction η, calculate $V^-(z)$ and $V^+(z)$ as proposed in Lee et al., but then at the end we rescale $V^-(z)$ and $V^+(z)$ by the length of the directions we considered. Why is the rescaling necessary and why is it mentioned nowhere in the papers?

Set.seed()

Is our practice for setting seeds in the while loops in the simulation files ok? The function still is a bit of a mystery to both us.

Constant fraq

Would it be "fair" to compare Drysdale's p-values with Christoph's p-values, when having them at different fractions (i.e split-rates) to ensure Drysdale's $\hat{\beta}^{Carve}$ to exist?

Theoretical notes:

Notes to ourselves:

Conditioning on s:

I asked Filip on Friday how you actually compute things when you only want to condition on one sign pattern. Lee makes this clear on p. 15: "Conditioning on the signs means that we only have to compute the interval [V-s(z), V+s(z)] for the sign pattern s that was actually observed."

We see right under Theorem 5.3 in Lee, that V-s(z) and V+s(z) are defined through A=As and b=bs. And s influences the definitions of A1(M,s) and b1(M,s) respectively.

Since s is in {-1,1}^|M|, it's only defined for variables that are actually selected, so the computation of the signs is straightforward (I mention this, because we had some confusion with a similar thing in another paper where we had s in [-1,1]^|M| or sth like this)

Question: Which beta^hat are we actually using though to get the signs? A priori all of beta^carve, beta^POSI and beta^SPLIT seem at least viable

Thinking about it, I guess that since we are talking about M (i.e. M_A) all the time, it is probably beta^Split, which is also the beta we are working with in the code above. In fact, Filip already implemented it exactly like that above.

Multiple polyhedra:

Question: If we only have eta in R^nx1 for a single polyhedron and eta_M in R^nx|M| for the union of polyhedra: What eta_M do we actually use now when we additionally condition on the signs, to only have one polyhedron?

Definition of $m_j(x)$ in Lemma 3.1

Drysdale writes $m_j(x) = (x - \theta_x)/\sigma_x$. Since θ_x, σ_x aren't defined, I guess he means:

$$m_j(x) = (x - \theta_j)/\sigma_j$$

Changes Made

Paul Sunday, 24rd March:

- Moved the theoretical notes over from carve_linear to this markdown file
- Try whether we get reasonable values from the SNTN Cdf when putting in very "average" values ** For z=0, 1,-1 respectively, we got the values 1/2, 0.86, 0.13, which seems reasonable (not sure how much the standard deviation rules of the normal distribution still apply here)
- Added set.seed(42) to carve.linear to have replicability while debugging.
- Question: Are p-values of all 0 actually a problem? Isn't that exactly what we'd like when testing for betas, that are as big as the ones we get in our examples? - Let's compare the p-values for all 9 entries of our $\hat{\beta}^{Carve}$ ** For $\hat{\beta}_4^{Carve} = 139.116200$ we get: 0 ** For $\hat{\beta}_3^{Carve} = -7.379114$ we get: 1 ** Problem: When running the code for the Toeplitz example, we get $\hat{\beta}^{Carve} \in \mathbb{R}^9$, but when calculating the p-values, we only get 6. Where do the 3 values get lost? *** Answer - this doesn't happen, just seemed so,

because I ran it twice back to back and actually got differently sized βs due to the randomness of the Lasso.

- Division by 0 in sntn_cdf: ** This happens $\iff$ $\Phi(\delta) = \Phi(\omega)$. In theory this shouldn't happen, because $\Phi(\delta) = \Phi(\omega) \iff a = b$ with a, b being the truncation limits of the truncated normal and it wouldn't make sense for them to be equal. However for "big" values for a and b (Already for a>=6), in R $\phi(a) = 1$, therefore the division by 0 occurs. ** Remedy: Since in this case even in theory, i.e. without computational approximation to 1, $\Phi(\delta) - \Phi(\omega)$ would be very small, as a consequence the whole of F would be very big, i.e (almost) equal to 1. Therefore: We implemented an if clause that sets F(z) to 1, if $\Phi(\delta) = \Phi(\omega)$ ** However: In these cases it also tends to be that the numerator = 0, i.e $B_\rho(m_1(z), \delta) = B_\rho(m_1(z), \omega)$ because of the same reasons as above. Since we don't know which one of numerator and denominator is actually bigger in this case, we set the probability to 0 by hand, which results in the p-value being set to 1. While this is unsatisfactory, it is the more conservative decision. ** Up for discussion: Maybe leaving it as NA would actually be the best decision?

Paul Monday, 25th March:

- Started running the simulation studies as discussed with Filip yesterday in the file called "Power Studies Toeplitz". I used a Toeplitz design again, but with lower noise and more active variables ($s_0 = 15$)
- We saw immediately that under the "right" conditions, $\hat{\beta}^{Drysdale}_{Carve}$ has the anticipated issue of not being able to compute β^{Split} due to rank issues.
- Note: I only saw that the computation crashed, but I don't know with 100% certainty whether this actually was the issue. *TODO*: Implement STOP messages, which would confirm this.
- I then went on to use a 60-40 split instead, on which $\beta^{Drysdale}_{Carve}$ could then be computed again - as well as the respective p-values. I also calculated the p-values for Christophs carving function.
- Then I started creating a "Confusion matrix" for Type I & II error. So far I've only done this for Christophs $\hat{\beta}^{Carve}$ though.
- *TODO*: Do the same for Drysdale as well - should be quite straightforward I think
- *TODO*: If possible, maybe try running a simulation that does all of the above e.g. a 100 times to see some proper results as far as power is concerned. Note: Computing time might be an issue, since even running Christophs carve.lasso only once in this specific Toeplitz example with many active variables took about 1 minute.

Filip Monday, 25th March:

Started with some experiments around the robustness of our estimators. As carve.lasso gives alot of "whitening constraints not fulfilled" errors on most of the seeds that I've tried, my new idea is to use Christoph's multi.carve, but with parameters set as such that it corresponds to regular carving. I checked also whether the choice of seed in the carving_simulation file propagates through to the functions and this is indeed the case. Furthermore, it was interesting to see that the selection events are not the same when comparing sel.models from carve_C and the chosen indices from our own split_select inside of carve.linear. This suggests that we still do not perform all constraint checks the same as it should be done for carve.lasso, which maybe explains the not fulfilled whitening constraints when calling carve.lasso on our own selection event. So an idea would be to take the selection event imposed by multi.carve and use it on carve.linear. I adapted multi.carve to return beta and lambda from its selection event, as well as carve.linear to not perform its own selection, but get it as parameters as carve.lasso does. To match the dimension of pvalues from multi.carve I set carve.linear's output to have also length 200, with ones at all indices, which were already excluded from selection. Seed 41 and fraq 0.9 gives a singular matrix error for carve.linear, at this seed there are many selected variables, could be that we encounter here the problems mentioned by Christoph.

Filip Tuesday, 26th March:

- Implemented stop messages in carve.linear which appear if the moore penrose inverse is not well defined due to singularity of $X_{A,M_A}^T X_{A,M_A}$ or $X_{B,M_A}^T X_{B,M_A}$
- Added confusion matrix example for Drysdales p-values in the style of what Paul did with Christophs p-values.
- Had major problems with replicability of carve.linear. The two selection events are different when comparing the split.select output inside of "Power Study Toeplitz" and the split.select output from inside of carve.linear. To get similar results, the seed has to be set again inside of carve.linear, or maybe be passed as an argument for later automatization.
- Tried setting the seed again before calling carve.linear, but this gives 0 selected variables from Lasso
- carve.linear works perfectly fine when not executing the split and carve.lasso in "Power Study Toeplitz" first. This behaviour seems very weird
- For the first possible comparison of the p_vals_C and p_vals_D it worked to set a different seed before calling carve.linear. This solution is temporary and still needs more investigation
- TODO: We can discuss if it would help to perform the selection only once inside of "Power Study Toeplitz" and adapt carve.linear to get the selection informations passed as arguments as it is in carve.lasso. This seems to be the cleanest solution and gives the most fair comparison. I already did something like that in the branch "Filip fights whitening errors".

Paul Tuesday, 26th March:

- Created the plots, which show the the simulations that Filip implemented.

Further literature

- PDF Selective inference Lee: https://cran.r-project.org/web/packages/selectiveInference/selectiveInference.pdf