Multicarving

Paul Schlossmacher

2024-03-13

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:

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

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, 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

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 $\hat{\beta}_j$:

$$n_B * \beta^{split} \dots$$

###Regarding vlo/vup: When calculating the truncation limits vlo and vup, we tried to do it similarly to what Drysdale does in his code. Mainly, we take a normalized row of the moore penrose inverse of x.Ma together with the sign of beta_split as the direction eta, calculate vlo and vup as proposed in Lee et al., but then at the end we rescale vlo and vup as well as the variance of the TN distribution by the length of the directions we considered. Why is the rescaling necessary and why is it nowhere mentioned?

As of now, we get sntn_cdf values of 1, mainly due to very high mean_delta and mean_w values. Normalizing the data did not resolve this.

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 aplly here)
- 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 $\hat{\beta}_5^{Carve} = 131.16525$ we get: ** For $\hat{\beta}_8^{Carve} = -10.30085$ we get:

** 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?