Notes on brms 10/10/21

So I think I’m going to have to make a trade-off between the number of sims (i.e. draws from the densityGMMs; see resims notes doc) and the number of routes that are analyzed simultaneously in a single model.

What I have been **trying** to do is draw many (I started with 100, and have bargained myself down to 50 → 25 → 5 which is just getting silly) sims from the GMMs and then analyze all the routes at once in a model of the form:

Total\_energy ~ (timeperiod \* source) / matssname

This is AWESOME AND PERFECT at small scales but does not run at scale. I have tried all weekend and stan keeps hanging, iterations don’t converge, warnings about ESS and BSS. I haven’t run any models with more than 100 routes to completion. Time requirements are scaling in nonlinear ways, the time estimates that brms communicates are orders of magnitude off. Weirdly it’s not getting killed by out of memory, but it is timing out.

So one possibility is that if I increased the memory and did this as a one off it would actually run through this way. However, I do also have suspicions that this is an unreasonably/unusually big compute ask. I have 528 routes with 10 years, and I want to estimate a separate slopes, intercepts, and interaction terms for each route. Ideally I would also eventually like to incorporate year as a random variable. Usually I think you would not be fitting the nested interactions as a fixed effect, because in most applications I think those are thought of as nuisance variables to account for but not to explicitly model, especially not at the level of a *nested interaction*.

So one option might be to post in weecology slack - something asking if folks who have fit more big models intuitively think this is too big, or that if I keep increasing memory it will run with finite compute.

:wave: I was wondering if I could ask some folks who have more experiencing fitting big models with brms a quick gut-check of whether something I’m trying to do seems like it will be feasible if I throw enough memory at it, or if it (as I suspect) is going to exceed a realistic compute ask?

I’m trying to fit a model on some BBS routes of the form

response ~ (timeperiod \* source) / route

Where response is a number (total biomass or total energy use), timeperiod is categorical with 2 levels (begin and end), source is categorical with 2 levels (real or simulated), and route is the BBS route (n = 500). Each time period contains 5 years and there are observations of response for each year. On top of this, there are numerous draws (at least 25 and preferably closer to 100) for the whole dataset corresponding to repeated draws from a null model.

Currently I am trying to run smaller trials of this - recently using up to 100 routes and 50 null model draws and 2000 iterations - using 4 chains and 4 cores of 32 GB each. These trials run in about 4-5 hours, but with lots of complaints about ESS/BSS and suggesting I try more iterations. None of my attempts to include more routes have successfully run within the 12 hours I give them, and - more concerning - they seem to hang about 200-400 iterations in, and the time estimates brms produces are wild underestimates. It seems like it *might* be just a question of giving it more memory to unstick it, but I’m hesitant to keep throwing more memory at the problem without a reality check. What really creeps me out is, the time and memory required appears to be increasing nonlinearly as I increase the number of routes and sim draws (let alone increasing iterations to improve performance), and I’m still nowhere near the “full” model.

My intuition is that this model might be unusually unwieldy because of the large number of routes and null model draws and the need for the nested interaction (as opposed to a single random intercept or something - I get the sense that it’s not usually done, but it’s kind of the whole point here), and that even with a lot of memory and time it might be impractical to run as one single model. I’m happy to abandon this and use an alternative; the null model draws and route-level interaction are way more important to the overarching scientific goals than getting everything to run in a single model. It’s not worth getting extremely fancy, or waiting 2 weeks for something to run, or displacing other folks’ resource use just to get this jammed through. Before I totally give up on it, though, I thought I’d post and see if folks who have more experience fitting things of this magnitude agree that it’s reasonable to decide that this is Computationally Intractable and seek alternatives? I ask because I don’t use brms much and don’t completely trust my intuition with it.

Having written this I have talked myself into alternatives.

Specifically….

* So I’m using brms and this nested structure to account for multiple comparisons in a frequentist framework.
* But I actually think the *way* more important thing is to get estimates that incorporate between-sim variability and (fingers-crossed) year as a random factor, than to (essentially) avoid under-estimating p values a little bit by running each route individually.
* One thing you could do would be to run route-level models and compare the CIs doing it route-level vs. doing it all one with just a few routes. Or you could not do that.
* You could also include (as a supplement-style thing) a single run with all the routes together using a single estimate for each year (mean of the sims).
  + Running all the routes with a single sim 2000 iterations completed in about 8 hours with warnings about Rhat and bulk and tail ESS all suggesting more iterations. The estimates have really bananas error bars. I am not sure if this is because of hidden covariance (I doubt it), only running with 1 sim, or of the model’s failure to converge with just 2000 iterations. I suspect the last but I’m not sure, I’m also a little suspicious of the 1 sim