1 Introduction

I use the following data-generating process

$$y = x_1\beta_1 + x_2\beta_2 + \sqrt{\frac{df-2}{df}}\varepsilon, \qquad \varepsilon \sim t_{df}$$

The reason that the error term looks the way it does is that I want to be able to specify data-generating processes with differently shaped error term distributions (varying thiccnes, as the kids would say) while keeping the error term variance fixed.

I then compare the two models

$$y = \gamma x_1 + \varepsilon_1, \qquad \varepsilon_1 \sim N(0, \sigma_1^2)$$
$$y = \phi x_2 + \varepsilon_2, \qquad \varepsilon_2 \sim N(0, \sigma_2^2)$$

The two models I compare are thus misspecified in two ways: they only include one of the two relevant covariates and their error-terms are normally instead of t-distributed.

I use a standard NIG (normal inverse-Gamma) prior for (γ, σ_1^2) and (ϕ, σ_2^2).

1.1 Deriving the Bayes factor

Given that the prior of σ_i^2 is $IG(a_0, b_0)$, and the prior of the regression coefficient is $N(0, \sigma_i^2 \Omega_0)$, the marginal distribution will be

$$p(y) = \text{MVSt}_{2a_0}\left(0, \frac{b_0}{a_0}(I + X\Omega_0 X^{\mathsf{T}})\right).$$

Given the marginal likelihood, it is straight forward to calculate the (log) Bayes factor as the (log of) the ratio of two multivariate t distributions.

2 Setup

I choose a very flat prior for the variance for the two models, letting both models b_0 and a_0 be close to zero. In a similar manner I set $\Omega_0 = I$ (which will just be 1 since there is only one regression parameter in each model) for both models.

3 Calculating variances

The main focus of the paper is to calculate sampling variances analytically for regression models. Can we do this in this appendix exercise as well? If not, why not?

4 The simulation study

The atomic element (one observation) of the simulation study is the generation of one sample from the DGP (a parent sample) that is resampled `n_bss` times. For each bootstrap sample, the log Bayes factor is calculated. Each parent then gives one estimated sampling variance of the log Bayes factor.

The function corresponding to the atomic element is `gen_one_line` , which takes as input arguments

- A design matrix with two covariates and n_obs observations. This typically stays the same between different parents, but I also try different design matrices to make sure the results are not the result of some weird design matrix setup.

- A thiccness parameter, deg_f, that is used to generate the error terms. The error terms are unique for each parent, even when they follow the same distribution.

- n_bss which indicates the number of bootstrap replicates to use for each parent. Ideally, this number would be arbitrarily large. I started with $n\_bss = 10\ 000$, but this is to large. It is more important to try different settings and to be able to afford to generate a large amount of parents, so I will use $n\_bss=1000$ instead. This is what we use in the DSGE example anyways.

- $omega\_0\_1$ and $omega\_0\_2$ which are related to the β priors.

- a_0 and b_0 which are related to the σ^2 priors. Note that both of the models that I compare have the same prior for the variance.

The atomic operation, which generates one line in the output, does the following

- Creates a vector that is n_bss to temporarily store values from the bootstrap estimates in.

- Creates a parent dataset of n_obs observations by by adding noise to the true response, where the added noise is t-distributed with deg_f degrees of freedom, and is rescaled to have the same variance no matter what deg_f is.

- Loops n_bss times: draw a bootstrap sample from the data, calculate the log Bayes factor based on that sample, store the result in the temporary vector.

- Calculates the log Bayes factor based on the parents original data.

- Estimates the sampling variance of the log Bayes factor by calculating the variance of the temporary vector of log Bayes factors, *lbf.*

- Calculates p_rad, the percentage of resampled datasets where the log Bayes factor is greater than 5 in absolute value.

- Returns a row consisting of n_bss, n_obs, deg_f, lbf, var_lbf, p_rad.

This atomic operation is called with the function $generate\_multiple\_rows$ that specifies all the information needed to generate one row, together with the number of parents to create ($runs$). It returns a matrix with $runs$ rows and six columns, corresponding to the six elements returned by $generate\_one\_row$.