Prior predictive checks for a drift-diffusion model

Yongfu Liao

Jan 19, 2024

Using the boundary separation parameter α parameter as an example here, we present the hierarchical structure of our drift-diffusion model below:

$$
\begin{aligned}
\alpha &\sim exp(\mu_\alpha + \sigma_\alpha * z_\alpha) \\
\mu_\alpha &\sim \text{Normal}(m_1^\alpha, s_1^\alpha) \\
\sigma_\alpha &\sim \text{Normal}^+(m_2^\alpha, s_2^\alpha) \\
z_\alpha &\sim \text{Normal}(0, 1)
\end{aligned}
$$

Other parameters $(\beta, \tau, \delta_1, \delta_2)$ follow similarly, where there are also hyper-parameters μ_*, σ_*, and z_* that co-determine the prior distributions of the lower-level parameters. Our goal here is to pick m_1^*, s_1^*, m_2^*, and s_2^* such that each of the lower-level priors covers a reasonable range.

Picking distributions for μ_*, σ_*, and z_* (i.e., determining m_1^*, s_1^*, m_2^*, and s_2^*) could be quite difficult as variations in these hyper-parameters interact to co-determine the lower-level priors. Intuitions for setting non-hierarchical priors can result in unreasonably wide priors in cases where there are hierarchical structures in the parameters (like our model here).

The only solution to reliably arrive at good hyper-priors is interactive simulations. In such a scenario, we first pick some hyper-priors, use these hyper-priors to simulate lower-level priors as well as observed variables, and look at whether they fall within a reasonable range (by comparing it against your knowledge about the phenomena and/or previous studies). Often, we have to repeat for several rounds to arrive at good priors. The technical term for this is *prior predictive checks*[1].

```r
library(ggplot2)
set.seed(2024)

# Function factory for creating link functions
trans_func = function(lnk = \(x) x) {
    function(m1, s1, m2, s2) {
        N = 5e5
        x = rnorm(N, m1, s1) + stom::rtnorm(N, m2, s2) * rnorm(N)
        lnk(x)
    }
}
# Prior simulators given hyper-parameters
sim = list(
    alpha  = trans_func(exp),
    tau    = trans_func(exp),
    beta   = trans_func(stom::inv_logit),
    delta1 = trans_func(),
```

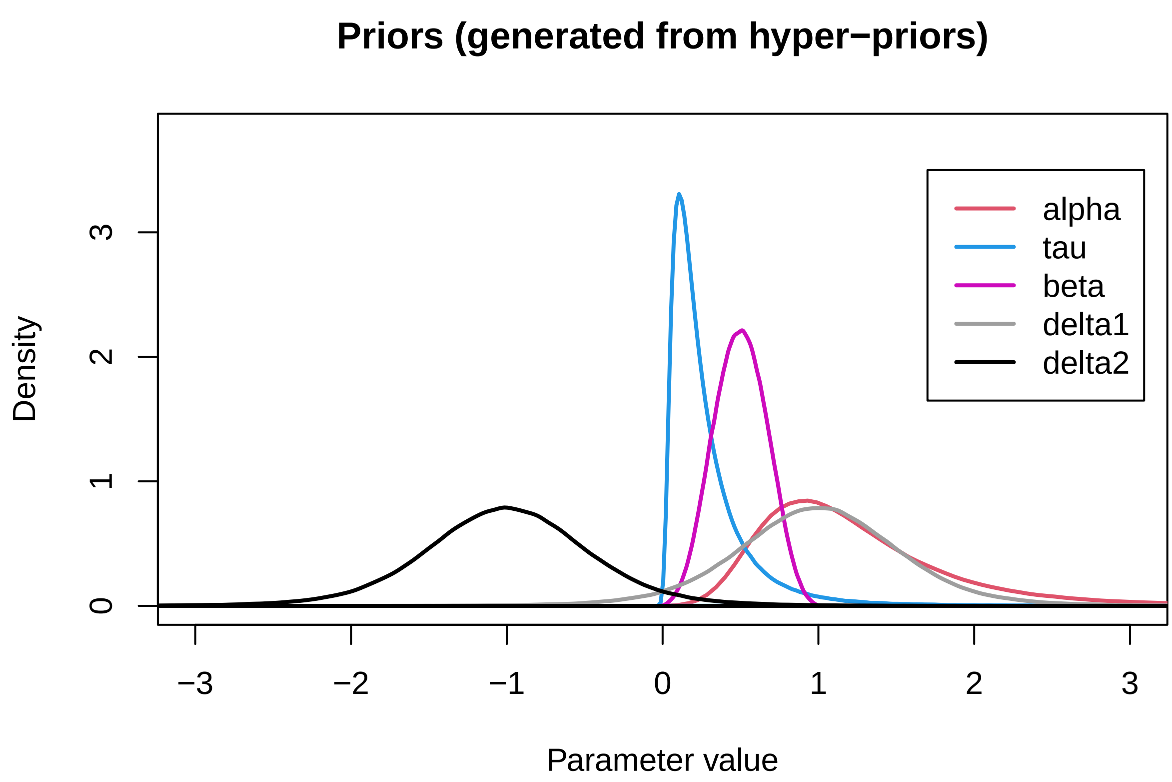
[1] https://mc-stan.org/docs/stan-users-guide/prior-predictive-checks.html

```
    delta2 = trans_func()
)
```

Prior distribution simulations

The code below documents the simulation of prior distributions based on a set of hyper-priors, which we have arrived at through multiple rounds of interactive simulations.

```
####   (Sanity check for hyper-parameters)   ####
a   = sim$alpha(m1=.1,s1=.4,m2=0,s2=.3)
t   = sim$tau(m1=-1.6,s1=.8,m2=0,s2=.2)
b   = sim$beta(  m1=0,s1=.6,m2=0,s2=.5)
d1  = sim$delta1(m1=1,s1=.4,m2=0,s2=.4)
d2  = sim$delta2(m1=-1,s1=.4,m2=0,s2=.4)
plot(1, type="n", xlim=c(-3,3), ylim=c(0,3.8),
     xlab="Parameter value", ylab="Density",
     main="Priors (generated from hyper-priors)")
polygon(density(a),  border = stom::col.alpha(2,1), lwd=2 )
polygon(density(t),  border = stom::col.alpha(4,1), lwd=2 )
polygon(density(b),  border = stom::col.alpha(6,1), lwd=2 )
polygon(density(d1), border = stom::col.alpha(8,1), lwd=2 )
polygon(density(d2), border = stom::col.alpha(9,1), lwd=2 )
legend(1.7,3.5,
       legend = stom::as_vec("alpha,tau,beta,delta1,delta2"),
       col = c(2,4,6,8,9),
       lwd = 2)
```



Prior predictive checks

The code below uses the above priors to simulate observations from a single subject based on our drift-diffusion model. The results are visualized as the reaction time distributions below.

```
sim_draws = function(n, a, t, b, d) {
    dt = list(
        q = vector("double", n),
        resp = vector("character", n)
    )
    for ( i in 1:n ) {
        y = RWiener::rwiener(1, alpha = a[i], tau = t[i], beta = b[i], delta = d[i])
        dt$q[i] = y$q
        dt$resp[i] = y$resp
    }
    return(data.frame(dt))
}

c1 = cbind( sim_draws(1000, a, t, b, d1), cond = "Stimulus 1" )
c2 = cbind( sim_draws(1000, a, t, b, d2), cond = "Stimulus 2" )
d = rbind(c1, c2)

ggplot(d) +
    geom_density(aes(q, fill=resp, color=resp), alpha=.1) +
    facet_grid(vars(cond)) +
    theme_bw() +
    labs(x = "RT", title = "Prior Predictive Check")
```