Prior predictive checks for a drift-diffusion model

Yongfu Liao

Jan 19, 2024

In the model here (using the α parameter as an example), we assume:

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

Other parameters $(\beta, \tau, \delta_1, \delta_2)$ follow as well, 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 parameters falls in 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 the variation in these hyper-parameters interact to influence the lower-level prior. Intuitions for setting non-hierarchical priors can result in unreasonably wide priors in cases where there are hierarchical structures in the parameters.

The only reliable solution to specify good hyper-priors is interactive simulations. We first pick some hyper-priors in order to simulate lower-level priors and observed variables. This provides feedback on the sanity of our hyper-priors such that we can then perform future rounds of simulations to improve the hyper-priors. The technical term for this is *prior predictive checks*[1].

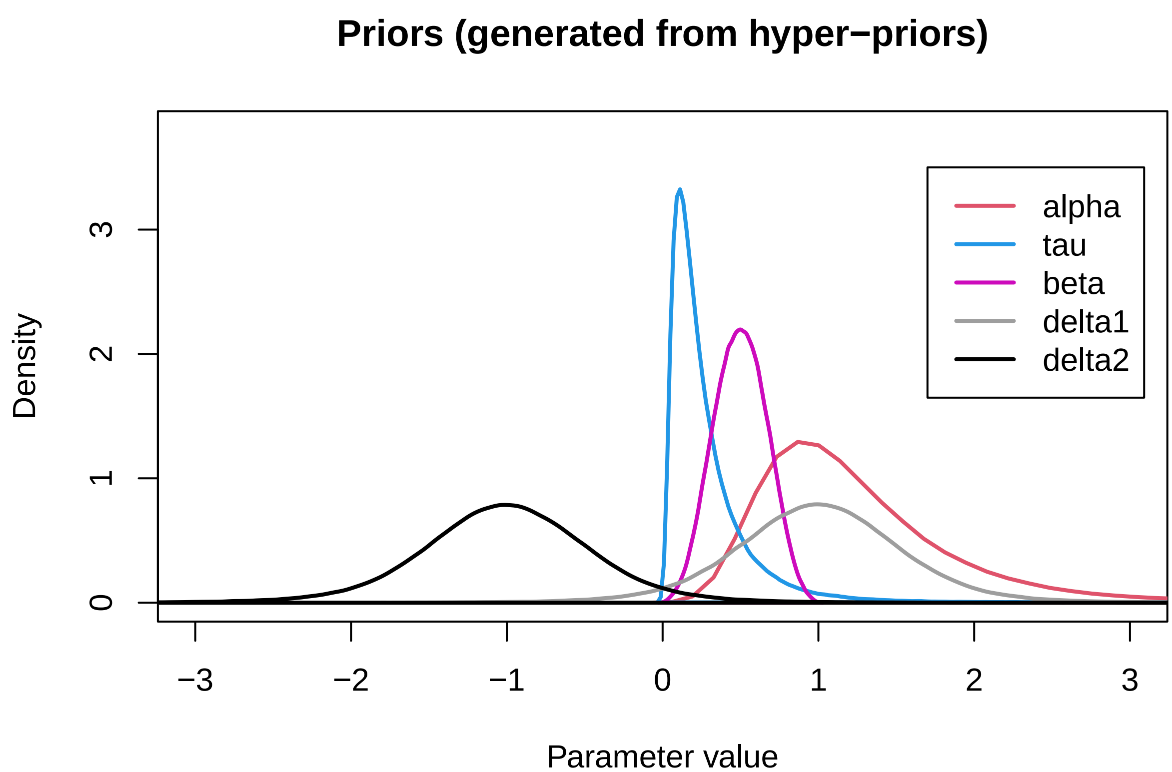
```r
library(stom)
library(ggplot2)
set.seed(2029)

# 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(),
    delta2 = trans_func()
)
```

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

Parameter prior distribution simulation

The code below documents the generation of prior distributions based on the final hyper-priors we have zoomed in on based on multiple rounds of 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 = col.alpha(2,1), lwd=2 )
polygon(density(t),  border = col.alpha(4,1), lwd=2 )
polygon(density(b),  border = col.alpha(6,1), lwd=2 )
polygon(density(d1), border = col.alpha(8,1), lwd=2 )
polygon(density(d2), border = col.alpha(9,1), lwd=2 )
legend(1.7,3.5,
       legend=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 based on the drift-diffusion model (for a single-subject). The results are plotted for visual inspections of the range in which the observations fall in.

```
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")
```