

Matrix normal sampling in Stan

Jordan S. Martin

6/15/2021

A simple demonstration of how to use matrix normal sampling to estimate a Kronecker product in Stan. The simulation generates bivariate normal responses with correlated phylogenetic effects (population $r = 0.5$, $\sigma^2 = 0.5$) and independent residuals.

```
#simulate scaling matrix
A = rethinking::rlkjcorr(1, 300, 1)

#simulate correlated phylogenetic effects
r_G = 0.5 #phylo correlation
v_G = 0.5 #phylo variance

G_cor <- matrix(c(1,r_G,r_G,1), nrow=2, ncol=2)
G_sd <- c(sqrt(v_G),sqrt(v_G))
G <- diag(G_sd) %*% G_cor %*% diag(G_sd)
Kron.prod <- G %x% A
P <- matrix(mvtnorm::rmvnorm(1, mean=rep(0,nrow(A)*2), sigma=Kron.prod), ncol = 2)
cor(P)

##           [,1]      [,2]
## [1,] 1.0000000 0.5234388
## [2,] 0.5234388 1.0000000

#Gaussian responses
v_res = 0.5 #residual variance (assume independent errors)
t1 = 0 + P[,1] + rnorm(nrow(A), 0, v_res)
t2 = 0 + P[,2] + rnorm(nrow(A), 0, v_res)
df = list(traits = cbind(t1,t2), A = A, n = nrow(A), p = 2)
```

```

#stan model
write(
  "data{
    int n; //number of species
    int p; //number of traits
    vector[p] traits[n]; //gaussian traits
    cov_matrix[n] A; //phylo vcv
  }
  transformed data{
    cholesky_factor_corr[n] L_A = cholesky_decompose(A);
  }
  parameters{
    matrix[n, p] z_phylo; //phylo standard normal deviates
    vector<lower=0>[p] sd_phylo; //phylo standard deviations
    cholesky_factor_corr[p] cor_phylo; //phylo correlations

    vector<lower=0>[p] sd_res; //residual standard deviation
    cholesky_factor_corr[p] cor_res; //residual correlation
  }
  transformed parameters{
    matrix[n, p] u_phylo; //scaled random effects
    //matrix normal parameterization of Kronecker product between G and A
    u_phylo = L_A * z_phylo * diag_pre_multiply(sd_phylo, cor_phylo)' ;
  }
  model{
    vector[p] mu[n];
    for (i in 1:n) {
      mu[i] = u_phylo[i]'; //assume z-scores with intercept = 0
    }

    traits ~ multi_normal_cholesky(mu, diag_pre_multiply(sd_res, cor_res) );

    to_vector(z_phylo) ~ std_normal();
    sd_phylo ~ exponential(1);
    cor_phylo ~ lkj_corr_cholesky(1);

    sd_res ~ exponential(1);
    cor_res ~ lkj_corr_cholesky(1);
  }
  generated quantities{
    corr_matrix[p] R_phylo = cor_phylo * cor_phylo';
    vector[p] v_phylo = sd_phylo .* sd_phylo;
    vector[p] v_res = sd_res .* sd_res;
  }",
  "m1.stan")

```

```
m1 = rstan::stan_model("m1.stan")

#estimate models
options(mc.cores = parallel::detectCores())
start_time <- Sys.time() #time model
mod <- rstan::sampling(m1, data=df, init = 0)
Sys.time() - start_time

## Time difference of 1.324435 mins

#results
post = rstan::extract(mod)
median(post$R_phylo[,2,1]) #phylo correlation

## [1] 0.5160415

apply(post$v_phylo, 2, median) #phylo variances

## [1] 0.4759387 0.6576377
```