A Bayesian multi-layered model to predict mechanisms, types, and severity of drug-induced liver injury.

Semenova Elizaveta, Williams Dominic, Lazic Stanley

Data Science and Quantitative Biology group, AstraZeneca, Cambridge UK

Data

- $N = 96$ - number of observations
- $N_0 = 21$ - number of covariates and their interactions
- $N_1 = 2$ - number of nodes in layer 1
- $N_2 = 2$ - number of nodes in layer 2
- $X[N, N_0]$ - design matrix (with interactions) of continuous covariates
- $y_1[N, N_1]$ - observed (binary) data of the first layer
- $y_2[N, N_2]$ - observed (binary) data of the second layer
- $y_3[N]$ - observed (ordered) severity class: 1 (safe), 2 (unsafe) or 3 (very unsafe)
- $c_{\text{max}}[N]$ - additional covariate to predict severity

Model formulation

Priors:

$$
\begin{aligned}
b_1 &\sim \text{Normal}(0, 5) & \beta_1 &\sim \text{Normal}(0, \tau_1 \lambda_1),\\
b_2 &\sim \text{Normal}(0, 5) & \tau_1 &= \tau_{\text{std}} * \tau_{01}, \tau_{\text{std}} \sim C^+(0, 1), \tau_{01} = 0.05,\\
\sigma_2 &\sim \text{Normal}^+(0, 1) & \lambda_1 &\sim C^+(0, 1)\\
\sigma_3 &\sim \text{Normal}^+(0, 1) & \beta_2 &\sim \text{Normal}(0, \sigma_2)\\
c &\sim \text{Normal}(0, 20) & \beta_3 &\sim \text{Normal}(0, \sigma_3)
\end{aligned}
$$

Latent linear predictors:

$$
\begin{aligned}
\eta_1 &= X\beta_1 + b_1\\
\eta_2 &= \tilde{\eta}_1 \beta_2 + b_2\\
\eta_3 &= (\tilde{\eta}_2, c_{\text{max}})\beta_3
\end{aligned}
$$

Likelihood:

$$
\begin{aligned}
y_1 &\sim \mathrm{BernoulliLogit}(\eta_1)\\
y_2 &\sim \mathrm{BernoulliLogit}(\eta_2)\\
y_3 &\sim \mathrm{OrderedLogistic}(\eta_3, c)
\end{aligned}
$$

The operation $\tilde{\eta}$ expands the matrix of predictors η into the matrix of predictors and their pairwise products to account for interactions; $(\tilde{\eta}_2, c_{\max})$ denotes horizontal stacking of the matrix $\tilde{\eta}_2$ and the vector $c_{\max}$.

Stan code

```
functions {

 // expand a vector of predictors of length K into a vector of predictors and their interactions
 vector v_long(int K, vector v_short){
    vector[K*(K+1)/2] v_long;
    v_long[1:K] = v_short;

    for (i in 1:(K-1)){
      v_long[((2*K -i + 1)*i/2 + 1): ((2*K -i + 1)*i/2 + K-i)] = v_short[i] * v_short[(i+1):K];
    }
    return v_long;
  }
}

data{
  int<lower=1, upper=3> K;              // number of classes
  int<lower=1> N;                       // number of observations
  int<lower=1> N0;                      // number of assays and their interactions
  int<lower=1> N1;                      // number of mechanisms
  int<lower=1> N2;                      // number of injury types

  matrix[N, N0] X;                      // assay design matrix
  vector[N] cmax;                       // additional predictor for severity

  int<lower=0, upper=1> y1[N, N1];      // observed mechanisms
  int<lower=0, upper=1> y2[N, N2];      // observed types
  int<lower=1, upper=K> y3[N];          // observed severity
}

transformed data{
  real<lower=0> tau01=0.05;
}
parameters{
```

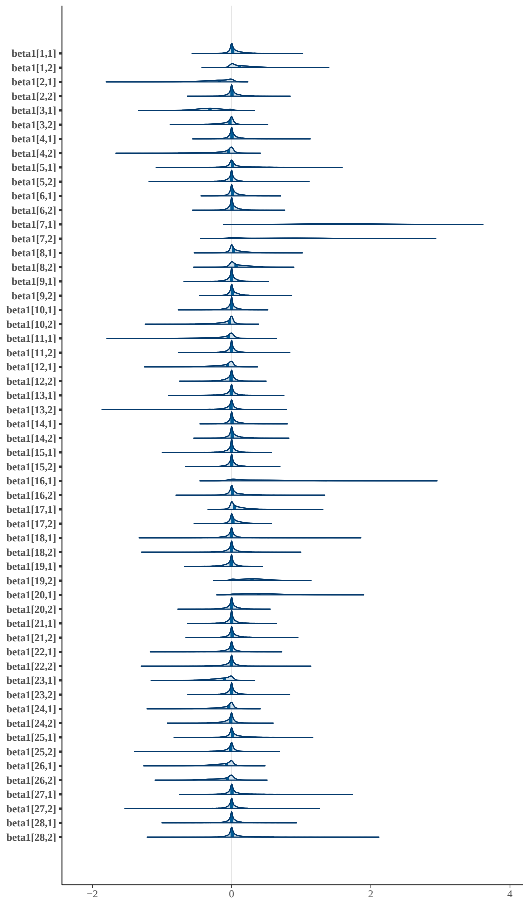
```
  row_vector[N1] b1;
  matrix[N0, N1] z1;

  row_vector[N2]           b2;
  matrix[N1*(N1+1)/2, N2] z2;

  vector[N2*(N2+1)/2 + 1] z3;

  ordered[K-1] cutpoints;

  real<lower=0> sigma2;
  real<lower=0> sigma3;

  matrix<lower=0>[N0, N1] lambda1;
  real<lower=0> tau_std;
}

transformed parameters{
  matrix[N0, N1]           beta1;
  matrix[N1*(N1+1)/2, N2] beta2 = z2 * sigma2;
  vector[N2*(N2+1)/2 + 1] beta3 = z3 * sigma3;
  matrix[N, N1]            b1_mat = rep_matrix(b1, N);
  matrix[N,N1]             eta1;
  matrix[N,N2]             eta2;
  vector[N]                eta3;
  real<lower=0>            tau1= tau_std * tau01;

  for (j in 1:N1){
    for (i in 1:N0){
        beta1[i,j] = z1[i,j] * tau1 * lambda1[i,j];
    }
  }

  eta1 = X * beta1 + b1_mat;

  {
     vector[N1] eta1_i_s;
     vector[N1*(N1+1)/2] eta1_i_l;
     vector[N2] eta2_i_s;
     vector[N2*(N2+1)/2] eta2_i_l;

     for (i in 1:N){
       eta1_i_s = to_vector(eta1[i,]);
       eta1_i_l = v_long(N1, eta1_i_s);
       eta2[i,] = b2 + eta1_i_l' * beta2;
       eta2_i_s = eta2[i,]';
       eta2_i_l = v_long(N2, eta2_i_s);
       eta3[i]  = append_col(eta2_i_l',  cmax[i]) * beta3;
```

```
      }
  }


}

model{

  // priors
  b1 ~ normal(0, 5);
  b2 ~ normal(0, 5);

  sigma2 ~ normal(0, 1);
  sigma3 ~ normal(0, 1);

  to_array_1d(lambda1) ~ cauchy(0,1);

  tau_std  ~ cauchy(0, 1);


  to_array_1d(z1) ~ normal(0, 1);
  to_array_1d(z2) ~ normal(0, 1);
  z3 ~ normal(0, 1);

  cutpoints ~ normal(0, 20);

  // likelihood
  to_array_1d(y1) ~ bernoulli_logit(to_array_1d(eta1'));
  to_array_1d(y2) ~ bernoulli_logit(to_array_1d(eta2'));
  for (i in 1:N){
   y3[i] ~ ordered_logistic(eta3[i], cutpoints);
  }

}

generated quantities{
  vector[N] log_lik;          // Log likelihood for WAIC / loo
  real y1_pred[N, N1];  // values for predictions
  real y2_pred[N, N2];
  real y3_pred[N];

  for (i in 1:N){

    //loglik
    log_lik[i] = ordered_logistic_lpmf(y3[i] | eta3[i], cutpoints);

    for (j in 1:N2){
      log_lik[i] = log_lik[i] + bernoulli_logit_lpmf(y2[i,j] | eta2[i,j]);
```

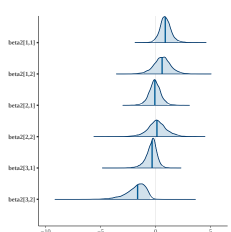
```
    }

    for (j in 1:N1){
      log_lik[i] = log_lik[i] + bernoulli_logit_lpmf(y1[i,j] | eta1[i,j]);
    }

     //predictions
     y3_pred[i] = ordered_logistic_rng(eta3[i], cutpoints);

     for (j in 1:N2){
        y2_pred[i, j] = bernoulli_logit_rng(eta2[i,j]);
     }

     for (j in 1:N1){
        y1_pred[i, j] = bernoulli_logit_rng(eta1[i,j]);
      }
  }

}
```

dili.stan

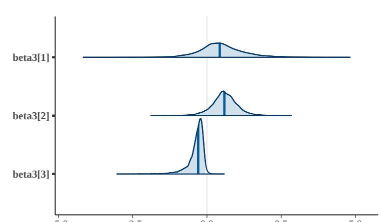
Posterior distributions with medians and 95% Bayesian credible intervals



Coefficients β_1 and the shrinkage effect.



Coefficients β_2



Coefficients β_3