Online Appendix
to
Covariance Regression Models for Studying Treatment Effect Heterogeneity Across One or More Outcomes: Understanding How Treatments Shape Inequality

Deirdre Bloome
University of Michigan

Daniel Schrage
University of Southern California

Sociological Methods and Research

Appendix A: Hoff and Niu (2012) Covariance Regression Model

Using a random-effects representation, the covariance regression model for the observed data can be written as

$$\boldsymbol{Y}_i = \boldsymbol{\mu}_{X_i} + \lambda_i \times \boldsymbol{B}\boldsymbol{X}_i + \boldsymbol{\epsilon}_i, \tag{A.1}$$

where $\boldsymbol{Y}_i$ is the multivariate response vector, $\boldsymbol{X}_i$ is a vector of predictor variables, $\boldsymbol{\mu}_{X_i}$ is a vector of conditional means for the responses given the predictors, $\boldsymbol{\epsilon}_i$ is the vector of residuals, and, crucially for the random-effects representation, λ_i describes additional variability beyond that captured in $\boldsymbol{\epsilon}_i$. $\boldsymbol{B}$ is a matrix; each row describes how the additional variability in the corresponding response variable is related to the predictor variables. Using standard assumptions of random-effects models,[1] this equation implies that the covariance matrix of $\boldsymbol{Y}_i$, given predictors $\boldsymbol{X}_i$, is

$$\boldsymbol{\Sigma}_{X_i} = \boldsymbol{\Psi} + \boldsymbol{B}\boldsymbol{X}_i\boldsymbol{X}_i^T\boldsymbol{B}^T, \tag{A.2}$$

where $\boldsymbol{\Psi}$ is the baseline covariance matrix of the residuals $\boldsymbol{\epsilon}_i$. The elements of $\boldsymbol{B}$ indicate how heteroskedasticity in the response vector varies across groups. For added flexibility, the model can be expanded in rank to allow somewhat more freedom in the relationship between the variances and covariances. A rank-2 model, for example, adds an additional set of random effects akin to λ_i and an additional matrix of parameters akin to $\boldsymbol{B}$; further rank expansion is accomplished in this way by adding more random effects and more sets of covariance parameters.[2]

Reference: Hoff, Peter, and Xiaoyue Niu. 2012. "A Covariance Regression Model." *Statistica Sinica* 22: 729–753.

[1] These assumptions include that the expectations of $\boldsymbol{\epsilon}_i$, λ_i, and $\lambda_i\boldsymbol{\epsilon}_i$ all equal zero, and the variance of λ_i is one. For estimation, we further assume that $\boldsymbol{\epsilon}_i$ is distributed multivariate normal and λ_i also follows a normal distribution.

[2] In practice, this increases the number of parameters and adds additional non-identifiability, which makes model fitting more difficult.

Appendix B: Stan Implementation of Covariance Regression

The Stan[3] code below implements our model, with two simplifications for clarity of presentation. First, the model fits data with only two outcomes. (All of the extra work needed to fit more than two outcomes is about efficiently managing the additional δ parameters, which distracts from the core of the model.) Second, this implementation follows the earlier model presentation in restricting the predictors $\boldsymbol{X}$ to be the same for the mean, variance, and correlation models rather than allowing different predictors for each, as we did in our empirical application. The results reported in this paper use a more general model that relaxes these two restrictions, and code for that model is available upon request.

[3] See `http://mc-stan.org` to download Stan and for documentation of the modeling language.

```
// Stan model for covariance regression
data {
  int<lower=1> n_obs; // # of obs
  int<lower=1> n_predictors; // # of params (including intercept)
  int<lower=2> n_outcomes; // p
  row_vector[n_outcomes] y[n_obs];
  row_vector[n_predictors] x[n_obs];
}
parameters {
  matrix[n_predictors,n_outcomes] beta; // mean coefficients
  matrix[n_predictors,n_outcomes] gamma; // variance coefficients
  vector[n_predictors] delta; // correlation coefficients
}
transformed parameters {
  cholesky_factor_cov[n_outcomes] Sigma_chol[n_obs];
  {
    matrix[n_outcomes,n_outcomes] Sigma;
    for(i in 1:n_obs) {
      Sigma[1,1] = exp(x[i]*col(gamma,1));
      Sigma[2,2] = exp(x[i]*col(gamma,2));
      Sigma[1,2] = sqrt(Sigma[1,1])
                   * sqrt(Sigma[2,2])
                   * ((2*inv_logit(x[i]*delta)) - 1);
      Sigma[2,1] = Sigma[1,2];
      Sigma_chol[i] = cholesky_decompose(Sigma);
    }
  }
}
model {
  // Weak priors on model coefficients.
  to_vector(beta) ~ cauchy(0, 1);
  to_vector(gamma) ~ cauchy(0, 1);
  delta ~ cauchy(0, 1);

  // Multivariate normal likelihood.
  // In multi_normal(), Sigma is not vectorized, so loop over rows of y.
  for(i in 1:n_obs) {
    y[i] ~ multi_normal_cholesky((x[i] * beta), Sigma_chol[i]);
  }
}
```