Multilevel linear models; covariance among parameters

ISTA 410 / INFO 510: Bayesian Modeling and Inference

U. of Arizona School of Information

October 20, 2021

Outline

Last time:

- Interactions
- Generalized linear models

Today:

- One more GLM
- Multilevel regression / hierarchical linear models
- Varying intercepts and slopes
- Covarying parameters

Binomial regression

Logistic regression

Most familiar GLM: binomial regression (aka logistic regression)

- Binomial outcome, logit link
- Underlying parameter

$$
\begin{aligned}
y_i &\sim \mathrm{Binomial}(p, n_i) \\
\mathrm{logit}(p) &= \alpha + \boldsymbol{\beta} \cdot \mathrm{x}
\end{aligned}
$$

- In PyMC3, use `pm.math.invlogit`

Funding data for NWO grants

Example: funding data for NWO grants

- NWO (Dutch research council) awards funding to researchers in many fields
- We have a data set of application and approval counts for NWO grants, stratified by field and by applicant gender (in this data set, male or female)
- Research question: is there bias toward male applicants?

A simple model

A model:

$$
\begin{aligned}
y_i &\sim \mathrm{Binomial}(n_i, p_i) \\
\mathrm{logit}(p_i) &= \alpha_{\mathrm{gender}(i)} \\
\alpha &\sim \mathrm{Normal}(0, 2)
\end{aligned}
$$

Prior on α: quite vague, prefer log-odds between ± 4

A DAG

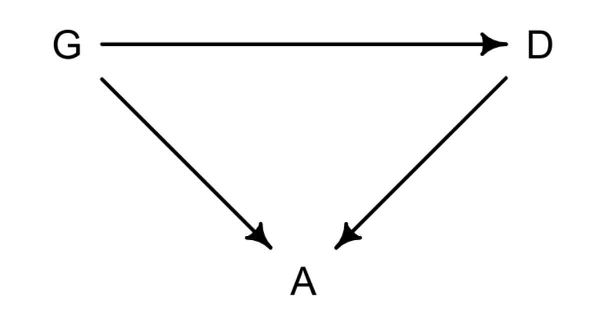
The computation suggests a noticeable gap between men and women: 3 percentage points on average, but with funding rates quite low, 3 percentage points is not so small.

But is this a direct causal effect, or mediated by an intermediate variable?

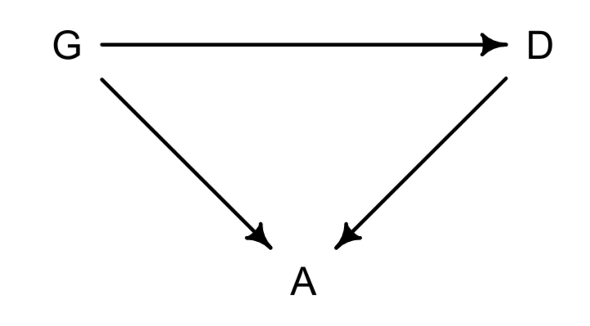
A DAG

The computation suggests a noticeable gap between men and women: 3 percentage points on average, but with funding rates quite low, 3 percentage points is not so small.

But is this a direct causal effect, or mediated by an intermediate variable?



A DAG



Two causal paths:

- Direct path $G \to A$
- Indirect path $G \to D \to A$

Previous model measured the two combined. Question about bias: is the direct effect nonzero?

A simple model

A model including discipline:

$$
\begin{aligned}
y_i &\sim \mathrm{Binomial}(n_i, p_i) \\
\mathrm{logit}(p_i) &= \alpha_{\mathrm{gender}(i)} + \beta_{\mathrm{discipline}(i)} \\
\alpha_j &\sim \mathrm{Normal}(0, 2) \\
\beta_j &\sim \mathrm{Normal}(0, 1)
\end{aligned}
$$

A multilevel model

Since the number of applications varies widely across disciplines (almost a factor of 10 from the least (physics) to most (social sciences)), we can also introduce partial pooling:

$$
\begin{aligned}
y_i &\sim \mathrm{Binomial}(n_i, p_i) \\
\mathrm{logit}(p_i) &= \alpha_{\mathrm{gender}(i)} + \beta_{\mathrm{discipline}(i)} \\
\alpha_j &\sim \mathrm{Normal}(0, 2) \\
\beta_j &\sim \mathrm{Normal}(0, \tau) \\
\tau &\sim \mathrm{HalfCauchy}(5)
\end{aligned}
$$

Hierarchical linear models

Recap: linear regression as a Bayesian model

Remember the basic framework we had for a linear model in the Bayesian setting:

$$
\begin{aligned}
y_i &\sim \mathrm{Normal}(\mu_i, \sigma) \\
\mu_i &= \alpha + \beta \cdot \mathsf{x_i} \\
\sigma &\sim \mathrm{HalfCauchy}(\phi) \\
\beta_i &\sim \mathrm{Normal}(0, \sigma_\beta) \\
\alpha &\sim \mathrm{Normal}(0, \sigma_\alpha)
\end{aligned}
$$

(different prior choices possible, of course!)

Recap: hierarchical models

Recall the idea of a hierarchical model:

- Observations are grouped into clusters
- Model parameters for each group come from a prior distribution dependent on population-level *hyperparameters*
- Allows for "partial pooling"; clusters don't all have the same model parameters, but some information is shared across clusters
- Effect: shrinkage toward population parameters, especially for clusters with few observations

Example: ozone in Beijing

Data set: $\sim$ 5 years of air quality monitoring data from Beijing

- Weather properties: temperature, pressure, dewpoint, rain, wind speed
- Pollutants: ozone, sulfur dioxide, nitrous oxide, carbon monoxide, particulates
- Measurements collected hourly at 12 monitoring stations

Simple modeling task: model ozone (in μg/m^3, about twice ppb) as a function of temperature

Preprocessing: group measurements by date/station

Basic model

Exploratory plotting suggests:

- the log maximum daily ozone reading is linearly associated with high temperature
- use max instead of average to prevent daily temporal associations from contributing

Preliminary model:

$$\begin{aligned} \log O_3 &\sim \mathrm{Normal}(\theta, \sigma) \\ \theta &= \alpha + \beta_T T_{\max} \\ \alpha &\sim \mathrm{Normal}(0, 3) \\ \beta &\sim \mathrm{Normal}(0, 1) \end{aligned}$$

Prior predictive reasonableness

Hang on, let's check those priors:

$$\alpha \sim \mathrm{Normal}(0, 3)$$
$$\beta \sim \mathrm{Normal}(0, 1)$$

Prior predictive reasonableness

Hang on, let's check those priors:

$$\alpha \sim \mathrm{Normal}(0, 3)$$
$$\beta \sim \mathrm{Normal}(0, 1)$$

If $\alpha = 3, \beta = 1$, then on a 30 degree day we get $\log O_3 \approx 33$; meaning about 100 trillion ppb.

A really bad ozone day might be a couple hundred ppb (about $\log O_3 \approx 6$ or 7). Let's rein in these priors.

Prior predictive reasonableness

With new priors:

$$\begin{aligned}\log O_3 &\sim \mathrm{Normal}(\theta, \sigma)\\ \theta &= \alpha + \beta_T T_{\max}\\ \alpha &\sim \mathrm{Normal}(0, 2)\\ \beta &\sim \mathrm{Normal}(0, 0.1)\end{aligned}$$

Now a high estimate combined with a hot day gives us something more like $\log O_3 \approx 6$.

Let's proceed!

Model specification

In Python:

```
with pm.Model() as linear_model:
    alpha = pm.Normal('alpha', 0, 2)
    beta = pm.Normal('beta', 0, 0.1)

    # Model equation
    theta = alpha[dailies.dropna()['station_id']]
        + beta * dailies.dropna()['TEMP']

    # Likelihood
    y_ = pm.Normal('y', mu=theta, sigma = sigma,
         observed = dailies.dropna()['log_ozone'])
```

Model results

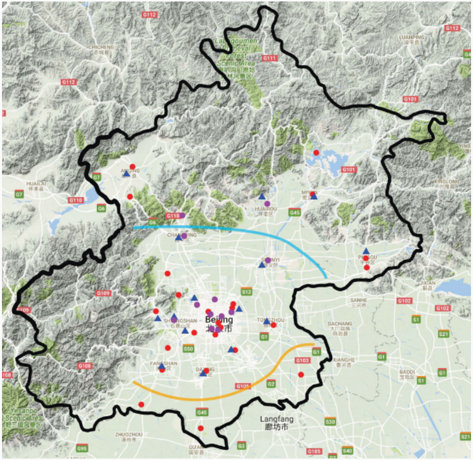
Here is a summary table for the preliminary model:

	mean	sd	hdi_3%	hdi_97%
alpha	3.454	0.009	3.436	3.471
beta	0.054	0.000	0.053	0.055
sigma	0.612	0.003	0.606	0.619

As expected:

- warmer days have more ozone
- specifically, change of 1 degree in high temperature associated with about a 5% increase in peak ozone concentration

Why multilevel model?



A map of Beijing. Purple dots are the 12 monitoring stations.

Why multilevel model?

We should expect some variation among geographic sites:

- Ozone source density may vary
- Topography influences local airflow patterns
- Sensor calibration differences

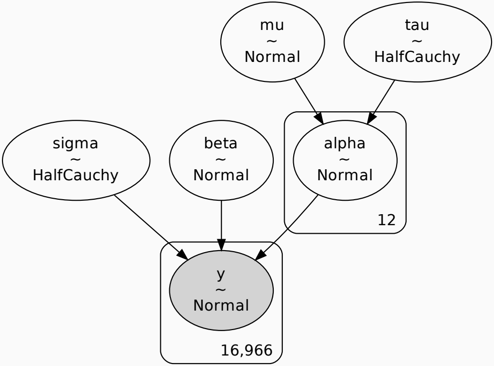
Simple extension of our model: allow varying intercepts across monitoring stations, to allow for each station to have a different "baseline" ozone level.

Varying-intercepts model

So let's extend the model:

$$
\begin{aligned}
\log O_3 &\sim \text{Normal}(\theta, \sigma) \\
\theta &= \alpha_j + \beta_T T_{\text{max}} \\
\beta &\sim \text{Normal}(0, 1) \\
\alpha_j &\sim \text{Normal}(\mu, \tau) \\
\mu &\sim \text{Normal}(0, 3) \\
\tau &\sim \text{HalfCauchy}(1)
\end{aligned}
$$

Varying-intercepts model



Model specification

```python
with pm.Model() as multilevel_model:
    # Hyperparameters
    mu = pm.Normal('mu', 0, 2)
    tau = pm.HalfCauchy('tau', 1)
    # Parameters
    sigma = pm.HalfCauchy('sigma', 1)
    alpha = pm.Normal('alpha', mu, tau, shape = 12)
    beta = pm.Normal('beta', 0, 0.1)
    # Model equation
    theta = alpha[dailies.dropna()['station_id']]
    + beta * dailies.dropna()['TEMP']
    # Likelihood
    y_ = pm.Normal('y', mu=theta, sigma = sigma,
        observed = dailies.dropna()['log_ozone'])
```

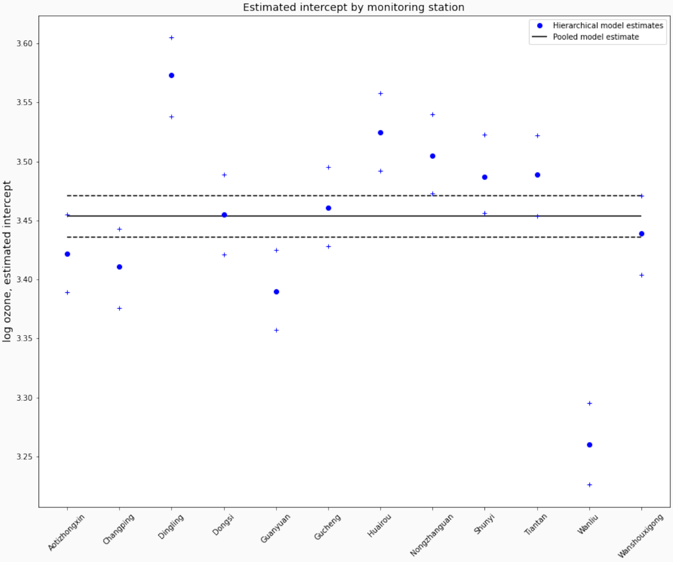
Model fitting

Results:

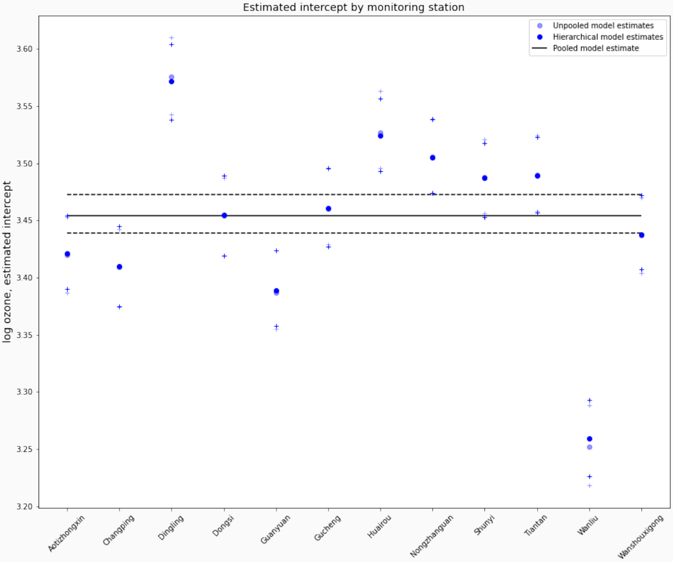
	mean	sd	hdi_3%	hdi_97%
mu	3.451	0.029	3.396	3.504
alpha[0]	3.422	0.018	3.389	3.455
alpha[1]	3.411	0.018	3.376	3.443
alpha[2]	3.573	0.018	3.538	3.605
alpha[3]	3.455	0.018	3.421	3.489
alpha[4]	3.390	0.018	3.357	3.425

- Hierarchical mean parameter similar to pooled intercept, but station intercepts vary

Model fitting



Model fitting



Model comparison

Comparing the models using PSIS-LOO:

	rank	loo	p_loo	d_loo	weight	se	dse	warning	loo_scale
multilevel	0	-15629.400655	17.039457	0.000000	0.969016	189.518973	0.00000	False	log
simple	1	-15757.707232	6.678447	128.306577	0.030984	190.474747	16.99114	False	log

- Multilevel model: better predictive score
- Also allows us to estimate which locations have elevated (Dingling) or depressed (Wanliu) baseline ozone

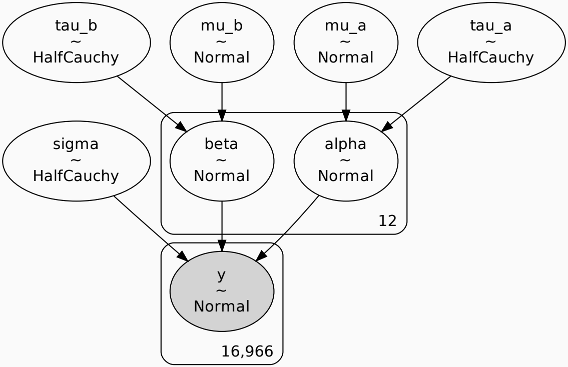
Varying slopes

Of course, once we have *varying intercepts*, it seems natural to also want *varying slopes*.

Extend the model again:

$$
\begin{aligned}
\log O_3 &\sim \mathrm{Normal}(\mu, \sigma) \\
\mu &= \alpha_j + \beta_{T,j} T_{\mathrm{max}} \\
\beta_{T,j} &\sim \mathrm{Normal}(\mu_\beta, \tau_\beta) \\
\alpha_j &\sim \mathrm{Normal}(\mu_\alpha, \tau_\beta) \\
\mu_\alpha &\sim \mathrm{Normal}(0, 2) \\
\tau_\alpha &\sim \mathrm{HalfCauchy}(1) \\
\mu_\beta &\sim \mathrm{Normal}(0, 0.1) \\
\tau_\beta &\sim \mathrm{HalfCauchy}(1)
\end{aligned}
$$

Varying slopes



Model comparison

Finally, we add the varying-slopes model to the model comparison:

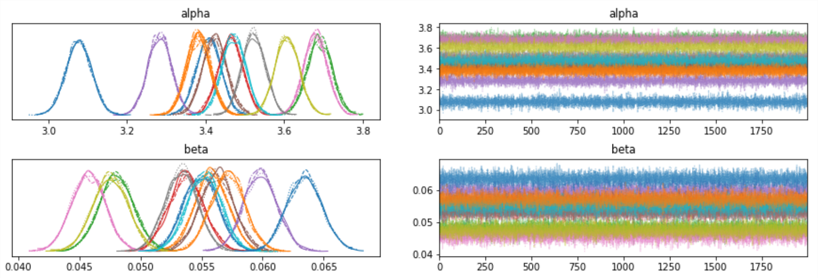
	rank	loo	p_loo	d_loo	weight	se	dse
varying slopes	0	-15557.545038	29.396342	0.000000	8.172186e-01	190.105625	0.000000
varying intercepts	1	-15629.400655	17.039457	71.855617	5.087248e-12	189.518973	13.373548
simple	2	-15757.707232	6.678447	200.162195	1.827814e-01	190.474747	26.581303

See a further improvement in model fit from varying slopes

Covarying parameters

Slopes and intercepts

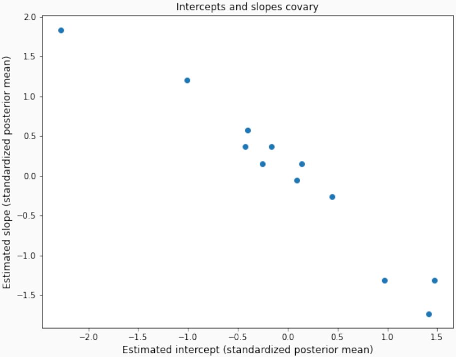
Here is part of the traceplot from the varying-slopes model:



Colors are the same. What do we see?

Covariance between parameters

- Intercepts vary across stations
- Slopes vary across stations
- Intercepts and slopes are correlated



A station with a high baseline ozone sees a smaller relative effect with temperature

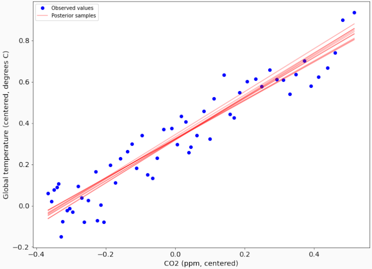
Covariance between parameters

Right now:

- slopes and intercepts are *a priori* independent
- we can see the negative association in the posterior, but the model cannot learn that pattern
- if we add a new monitoring station, it restarts with the same independence assumption:
  - the model knows something about typical intercepts
  - the model knows something about typical slopes
  - the model knows nothing about the relationship between them

Varying slopes and covariance

Covariance across parameter types is common for these models:



Each pair α, β represents a line

- They all have to stay close to the data
- If you change the slope you change the intercept

Covariance between parameters

In order to capture this effect, we need to explicitly include correlations in the model:

- Instead of thinking of 12 different intercepts and 12 different slopes, think of 12 intercept-slope pairs
- Right now, α_i, β_i are *a priori* normally distributed
- Replace the normal priors on α, β with multivariate normal

Multivariate normal distribution

Multivariate normal distribution:

- Generalizes normal distribution to produce vectors instead of scalars
- Specified by a vector of means and a covariance matrix
  - mean vector: contains mean of each component
  - covariance matrix: contains variance of each component, and covariance of each pair of components

Multivariate normal distribution

2×2 covariance matrix:

$$\Sigma = \begin{pmatrix} \sigma_a^2 & \sigma_a \sigma_b \rho_{ab} \\ \sigma_a \sigma_b \rho_{ab} & \sigma_b^2 \end{pmatrix}$$

- σ_a^2 – variance of a
- σ_b^2 – variance of b
- ρ_{ab} – correlation of a and b

The model

$$
\begin{aligned}
\log O_3 &\sim \mathrm{Normal}(\mu, \sigma) \\
\mu &= \alpha_j + \beta_j T_{\max} \\
\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} &\sim \mathrm{MVNormal}\left(\begin{pmatrix} \mu_\alpha \\ \mu_\beta \end{pmatrix}, \Sigma \right) \\
\Sigma &= \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \mathrm{R} \begin{pmatrix} \sigma_\alpha & 0 \\ 0 & \sigma_\beta \end{pmatrix} \\
\mu_\alpha &\sim \mathrm{Normal}(0, 2) \\
\mu_\beta &\sim \mathrm{Normal}(0, 0.1) \\
\sigma_\alpha &\sim \mathrm{HalfCauchy}(1) \\
\sigma_\beta &\sim \mathrm{HalfCauchy}(1) \\
\sigma &\sim \mathrm{HalfCauchy}(1) \\
\mathrm{R} &\sim LKJ(2)
\end{aligned}
$$

Multivariate normals and covariance matrices

Multivariate normal distribution

Multivariate normal:

- Generalization of the ordinary normal distribution to produce vectors
- Ordinary normal: parameterized by mean and variance
- MV normal: parameterized by mean vector and covariance matrix

Covariance matrix

m-by-m covariance matrix: for m components, m variables all varying together

- m standard deviations / variances
- $(m^2 - m)/2$ correlations / covariances
- total: $m(m+1)/2$ parameters

Covariance matrix

Covariance: measures how two random variables vary "together"

Covariance of two random variables:

$$\mathrm{Cov}\,(X, Y) = E[(X - E[X])(Y - E[Y])]$$

What's the covariance of X with itself?

$$\mathrm{Cov}\,(X, X) = E[(X - E[X])^2] = \sigma_X^2$$

Covariance matrix

Covariance matrix of $X_1, X_2, \ldots$:

$$\begin{pmatrix} \mathrm{Cov}(X_1, X_1) & \mathrm{Cov}(X_1, X_2) & \mathrm{Cov}(X_1, X_3) & \ldots \\ \mathrm{Cov}(X_2, X_1) & \mathrm{Cov}(X_2, X_2) & \mathrm{Cov}(X_2, X_3) & \ldots \\ \ldots & & & \end{pmatrix}$$

Covariance matrix

We can write this in terms of variances and correlations:

$$\begin{pmatrix} \sigma_1^2 & \sigma_1\sigma_2\rho_{12} & \sigma_1\sigma_3\rho_{13} & \dots \\ \sigma_1\sigma_2\rho_{12} & \sigma_2^2 & \sigma_2\sigma_3\rho_{23}\dots & \\ \dots & & & \end{pmatrix}$$

Example

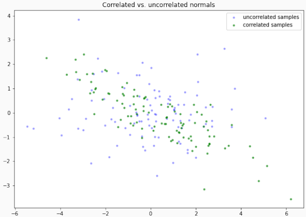
Say $\sigma_1^2 = 4, \sigma_2^2 = 1, \rho_{12} = -0.8$; then the covariance matrix is

$$\Sigma = \begin{pmatrix} 4 & -1.6 \\ -1.6 & 1 \end{pmatrix}$$

What does a sample from $\mathrm{MVNormal}(0, \Sigma)$ look like?

Example

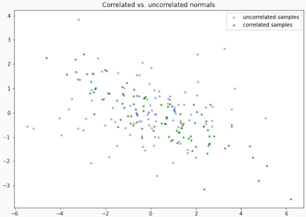
Compare to two independent normals with the same mean, variance:



What would the covariance matrix be for the independent samples?

Example

Compare to two independent normals with the same mean, variance:

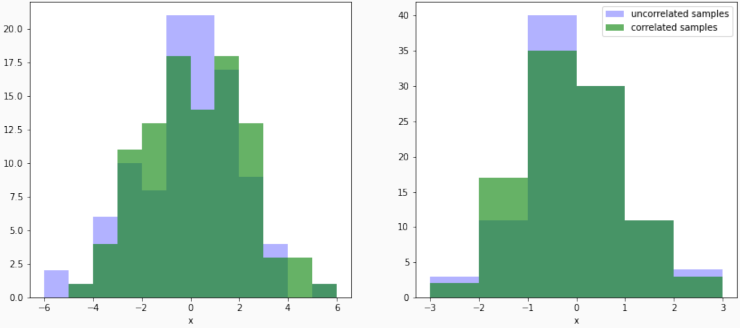


What would the covariance matrix be for the independent samples?

$$\Sigma = \begin{pmatrix} 4 & 0 \\ 0 & 1 \end{pmatrix}$$

Example

Compare the marginal distributions:



Can't tell apart! Marginals are $N(0,2)$ and $N(0,1)$ in both cases.

Priors for covariance matrices

Wishart distribution

Previously common: Wishart/inverse-Wishart distribution

Why not to use it:

- Often doesn't reflect our prior beliefs about correlations
- Can't decouple correlation and variance of individual variances
- Computationally hard for MCMC samplers
- Conjugacy not that big a deal anyway

Factoring the covariance matrix

Matrix multiplication interlude:

$$\begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots \\ b_{21} & b_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} = ?$$

Matrix multiplication

Grab 1st row and 1st column:

$$\begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots \\ b_{21} & b_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} = ?$$

Matrix multiplication

Grab 1st row and 1st column:

$$\begin{pmatrix} a_{11} & a_{12} & \dots \\ a_{21} & a_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} \begin{pmatrix} b_{11} & b_{12} & \dots \\ b_{21} & b_{22} & \dots \\ \vdots & \vdots & \end{pmatrix} = ?$$

Multiply componentwise and add:

$$\text{product entry} = a_{11}b_{11} + a_{12}b_{21} + a_{13}b_{31} + \dots$$

Matrix multiplication

2x2 example:

$$\begin{array}{cc} & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\ \begin{pmatrix} A & B \\ C & D \end{pmatrix} & \begin{pmatrix} & \\ & \end{pmatrix} \end{array}$$

Matrix multiplication

2x2 example:

$$\begin{array}{cc} & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\ \begin{pmatrix} A & B \\ C & D \end{pmatrix} & \begin{pmatrix} Aa + Bc & \\ & \end{pmatrix} \end{array}$$

Matrix multiplication

2x2 example:

$$
\begin{array}{cc}
 & \begin{pmatrix} a & b \\ c & d \end{pmatrix} \\
\begin{pmatrix} A & B \\ C & D \end{pmatrix} & \begin{pmatrix} Aa + Bc & Ab + Bd \\ Ca + Dc & Cb + Dd \end{pmatrix}
\end{array}
$$

2×2 **covariance matrix**

We can factor the covariance matrix:

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \begin{pmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

2×2 **covariance matrix**

We can factor the covariance matrix:

$$\begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix} \begin{pmatrix} 1 & \rho_{12} \\ \rho_{12} & 1 \end{pmatrix} \begin{pmatrix} \sigma_1 & 0 \\ 0 & \sigma_2 \end{pmatrix}$$

So we have two parts:

- diagonal matrix of standard deviations
- correlation matrix

Priors:

- Standard deviations are easy (Exp, HC, etc.)
- Correlations are a little trickier

Why do we need a fancy prior?

Need a prior for the correlation matrix R, and we can't just pick anything:

- Say we have 3 variables, x, y, z
- 3 pairwise correlations:
  - $\rho_{xy} = 0.9$
  - $\rho_{xz} = 0.8$
  - $\rho_{yz} = -0.9$
- What's wrong?

Why do we need a fancy prior?

Need a prior for the correlation matrix R, and we can't just pick anything:

- Say we have 3 variables, x, y, z
- 3 pairwise correlations:
  - $\rho_{xy} = 0.9$
  - $\rho_{xz} = 0.8$
  - $\rho_{yz} = -0.9$
- What's wrong?
- Not every matrix can be a correlation / covariance matrix!

LKJ disribution

LKJ distribution: a prior for correlation matrices.

- Named for Lewandowski, Kurowicka, and Joe (2009)
- Depends on a shape parameter η, which specifies difference from zero correlations
  - $\eta = 1$: uniform correlation matrices
  - $\eta > 1$: suppress big correlations
  - $\eta < 1$: prefer big correlations
- PDF:

$$p(\mathrm{R}|\eta) \propto (\det \mathrm{R})^{\eta-1}$$

- In our simple case:

$$p(\rho|\eta = 2) \propto 1 - \rho^2$$

LKJ distribution in PyMC3

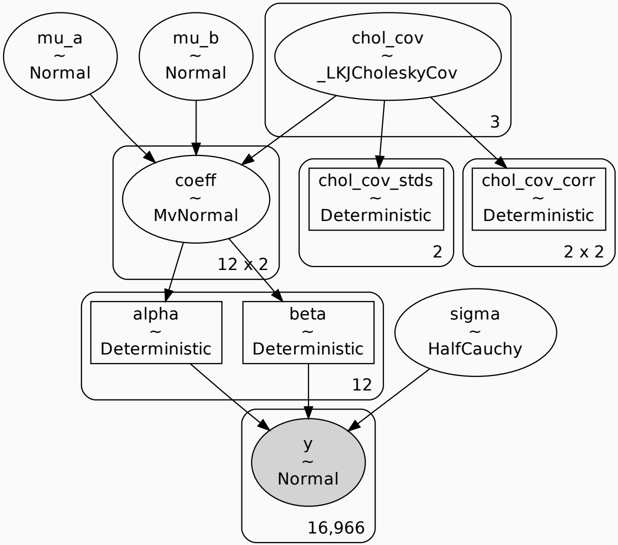
```
import theano.tensor as tt
with pm.Model() as cov_model:
    mu_a = pm.Normal('mu_a', 0, 2)
    mu_b = pm.Normal('mu_b', 0, 0.1)
    sigma = pm.HalfCauchy('sigma', 1)

    sd_dist = pm.HalfCauchy.dist(1)
    chol, corr, sd = pm.LKJCholeskyCov('chol_cov', eta = 2,
            n = 2, sd_dist = sd_dist, compute_corr = True)

    # tt.stack makes a vector; chol holds the covariance matrix
    coeff = pm.MvNormal('coeff', mu = tt.stack([mu_a, mu_b]),
                        chol=chol, shape=(12, 2))
    alpha = pm.Deterministic('alpha', coeff[:, 0])
    beta = pm.Deterministic('beta', coeff[:, 1])

    theta = alpha[data['station_id']] +
            beta[data['station_id']] * data['TEMP']
    y_ = pm.Normal('y', mu=theta, sigma = sigma, observed = data['log_ozone'])
```

Plate diagram



Results

When we run this model:

```
Sampling 4 chains for 500 tune and 500 draw iterations (2_000 + 2_000 draws total) took 172 seconds.
The acceptance probability does not match the target. It is 0.9894609716484106, but should be close t
The acceptance probability does not match the target. It is 0.9819508058560595, but should be close t
There were 457 divergences after tuning. Increase `target_accept` or reparameterize.
The acceptance probability does not match the target. It is 0.085127247008543, but should be close to
The acceptance probability does not match the target. It is 0.9893641325448185, but should be close t
The rhat statistic is larger than 1.4 for some parameters. The sampler did not converge.
The estimated number of effective samples is smaller than 200 for some parameters.
```

- Ugly stuff!
- But, we've run into this problem with hierarchical models before, and we know some tricks

Non-centered parameterization

Remember when we did the 8 schools model:

Bad:

$$\theta \sim \mathrm{Normal}(\mu, \sigma)$$

Good:

$$\theta = \mu + \sigma\eta$$

$$\eta \sim \mathrm{Normal}(0, 1)$$

- Essentially, modeling the standardized θs instead of modeling the θs directly:

$$x_i = \mu_i + \sigma z_i$$

- Mathematically equivalent, but much easier for the sampler

Non-centered parameterization

So if we standardized the usual hierarchical normal model using

$$\theta = \mu + \sigma\eta$$

what do we do for the multivariate version?

- μ is easy – just use the mean vector
- η is easy – just use a bivariate standard normal
- σ is the square root of the variance, so we should replace it with the square root of the covariance

Cholesky decomposition

When a matrix is symmetric and positive definite (e.g. a covariance matrix) it can be factored:

$$\Sigma = \mathsf{L}(\mathsf{L})^T$$

where L is a lower triangular matrix (hence L).

If z is a vector of independent standard normals, then Lz has the distribution of $\mathrm{MVNormal}(0, \Sigma)$ – exactly what we want!

$$\begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} = \begin{pmatrix} \alpha_j \\ \beta_j \end{pmatrix} + \mathsf{L} \begin{pmatrix} z_{\alpha,i} \\ z_{\beta,i} \end{pmatrix}$$

Good news: PyMC3 already gives us the Cholesky factor (because it's useful for numerical stability anyway)

Non-centered parameterization

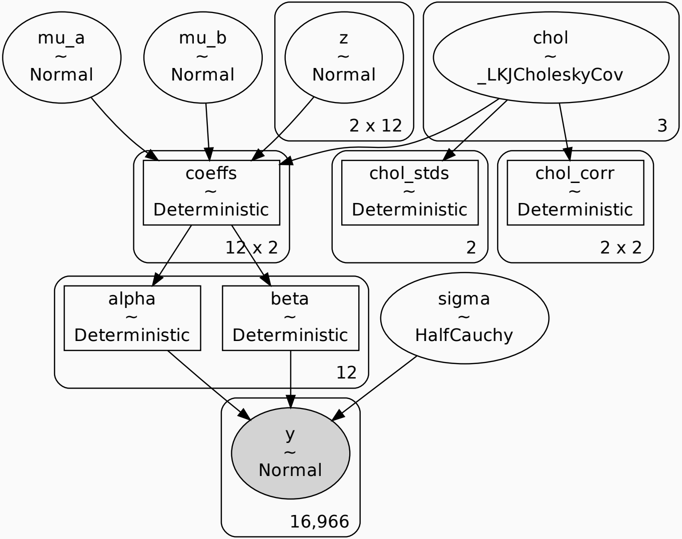
Finishing the non-centered parameterization:

```
import theano.tensor as tt
with pm.Model() as cov_model:
    # clipped out same stuff

    chol, corr, sd = pm.LKJCholeskyCov('chol', eta = 2, n = 2,
     sd_dist = sd_dist, compute_corr = True)
    # make standard normals
    z = pm.Normal('z', 0, 1, shape = (2, 12))
    # tt.dot is matrix multiplication
    coeffs = pm.Deterministic('coeffs',
             tt.stack([mu_a, mu_b]) + tt.dot(chol, z).T)

    # clipped out same stuff
```

Plate diagram

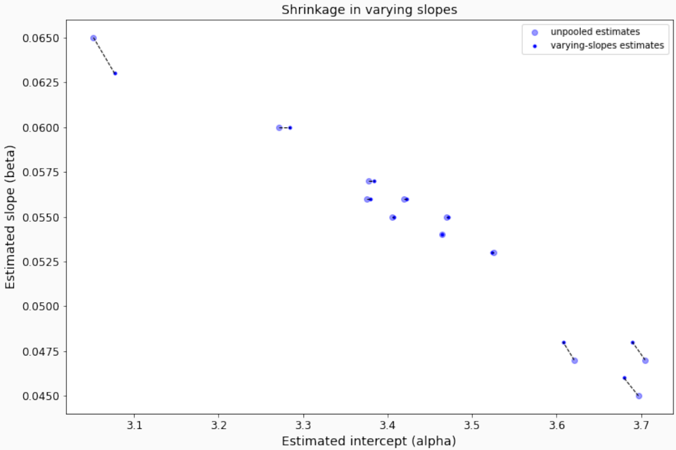


Results

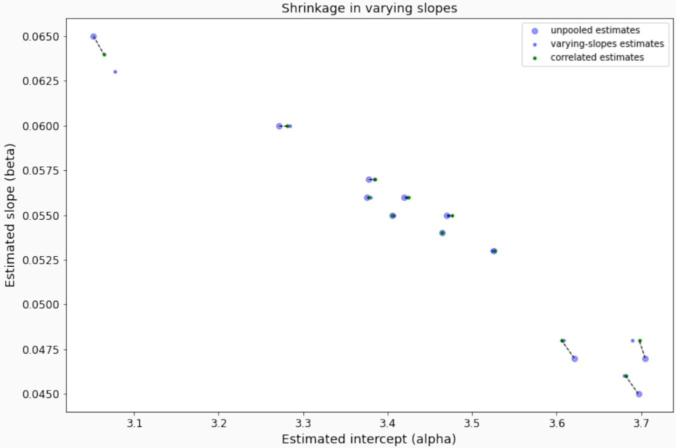
Result: no divergences, good Rhats, sampling takes about 2/3 the time

- α, β estimates similar to basic varying-slopes model
- Estimated correlation $\rho \approx -0.9$
- Two-dimensional shrinkage

Shrinkage with no correlations



Shrinkage with no correlations



What other correlations?

Are there other places we might expect to see correlations in this data set?

- Data come from air quality monitoring stations
- Monitoring stations are located in physical space
- Some are far apart, some are close together

Might reasonably expect that stations located close together are correlated

- Correlation between observations as a function of distance

Summary

- Using multivariate normal distributions, we can model parameters that are explicitly correlated
- Pool information across different types of parameters as well as different groups
- LKJ distribution gives a regularizing prior on correlation matrices

Next time:

- Spatial correlations and Gaussian processes
- No class Monday 10/25

Summary

Today:

- Hierarchical linear regression models
    - Varying intercepts
    - Varying slopes

Next week:

- Multivariate normal mechanics
- Priors for covariance matrices
- Gaussian processes