FlatPPL 0.1 / INTERMEDIATE

Hierarchical models

GROUP INTERCEPT

$$a_g = \mu_a + \tau z_g, \quad z_g \sim \mathcal{N}(0,1)$$

COMPLETE EXAMPLE

Grouped logistic regression

```
%%%
Noncentered grouped logistic regression.
%%%
flatppl_compat = "0.1"
G = 3
x_data = [-1.2, 0.4, 1.1, -0.3, 0.8, 2.0]
group_data = [1, 2, 3, 1, 2, 3]
y_data = [0, 1, 1, 0, 1, 1]

# Population parameters and group effects.
A mu_a ~ Normal(0, 1)
tau ~ LogNormal(0, 0.5)
z ~ iid(Normal(0, 1), G)
a = mu_a .+ tau .* z
beta ~ Normal(0, 2)

# Gather one group intercept per observation.
B eta = a[group_data] .+ beta .* x_data
probability = invlogit.(eta)
y ~ Bernoulli.(probability)

# Prior and likelihood in the latent coordinates.
C prior = lawof(record(mu_a = mu_a,
    tau = tau, z = z, beta = beta))
K = kernelof(record(y = y),
    mu_a, tau, z, beta)
L = likelihoodof(K, record(y = y_data))
posterior = bayesupdate(L, prior)

# Raise the intercept for group 2.
D point = record(mu_a = 0.0, tau = 1.0,
    z = [0.0, 1.0, 0.0], beta = 1.0)
log_L = logdensityof(L, point)
predict = functionof(probability,
    mu_a, tau, z, beta)
fitted = predict(point)
```

Use centered coordinates

```
a ~ iid(Normal(mu_a, tau), G)
```

Replace the draw and transform of `z` with this draw of `a`. Use `a` in the prior, kernel and evaluation point. Its conditional law stays the same.

Reuse the observation kernel

```
new_y = [0, 1, 0, 1, 1, 0]
L_new = likelihoodof(K, record(y = new_y))
```

Bind six new outcomes. `K` keeps the original groups, predictors and row order.

ANNOTATED LINES

A Noncentered intercepts 11–15

```
mu_a ~ Normal(0, 1)
tau ~ LogNormal(0, 0.5)
z ~ iid(Normal(0, 1), G)
a = mu_a .+ tau .* z
beta ~ Normal(0, 2)
```

Draw `G` independent standard Normals for `z`. Multiply each entry by `tau` and add `mu_a`. Conditional on these population parameters, each intercept has mean `mu_a` and standard deviation `tau`.

B Index the intercept by group 18

```
eta = a[group_data] .+ beta .* x_data
```

Change `z[2]` from 0 to 1 with $\mu_a = 0$ and $\tau = \beta = 1$. Two rows gain one unit of log odds. Applying `invlogit` gives their success probabilities.

Row	x_i	$z_2 = 0$	$z_2 = 1$
2	0.4	0.599	0.802
5	0.8	0.690	0.858

C The hierarchical likelihood 23–28

```
prior = lawof(record(mu_a = mu_a,
    tau = tau, z = z, beta = beta))
K = kernelof(record(y = y),
    mu_a, tau, z, beta)
L = likelihoodof(K, record(y = y_data))
posterior = bayesupdate(L, prior)
```

The prior has fields `mu_a`, `tau`, `z` and `beta`. Make all four variates inputs to `kernelof` to cut their prior dependencies. Then bind the observed outcomes with `likelihoodof`.

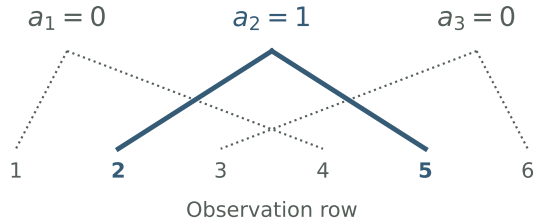
D Evaluate all six rows 31–36

```
point = record(mu_a = 0.0, tau = 1.0,
    z = [0.0, 1.0, 0.0], beta = 1.0)
log_L = logdensityof(L, point)
predict = functionof(probability,
    mu_a, tau, z, beta)
fitted = predict(point)
```

`predict(point)` returns the six probabilities in observation order. Set `z` to `[0, 0, 0]` on line 32 and evaluate again. Only entries 2 and 5 change. `log_L` scores the six observed outcomes at the same point.

INDEXING AND POOLING

Which rows use the new intercept?



At `point`, `a[group_data]` gives `[0, 1, 0, 0, 1, 0]`. Rows 2 and 5 share the intercept of 1. Their predictors differ.

Known errors for each group

```
theta ~ iid(Normal(mu_a, tau), G)
y_hat ~ Normal.(theta, [0.3, 0.2, 0.4])
```

As in eight schools, each group has an uncertain effect and a known standard error. Here the three errors correspond to groups 1, 2 and 3.

Pool success rates

```
phi ~ Beta(2, 2)
kappa = 10
rates ~ iid(Beta(phi*kappa, (1-phi)*kappa), G)
hits ~ Binomial.([8, 12, 10], rates)
```

`phi` sets the population mean. At fixed `phi`, increasing `kappa` concentrates the group rates around that mean. Supply a trial total and a success count for each group.

Recover the group intercepts

```
f_a = functionof(a, mu_a, tau, z)
intercepts = f_a(1, 2, [-1, 0, 1])
```

At $\mu_a = 1$ and $\tau = 2$, these latent values give the intercepts `[-1, 1, 3]`. Gather from this vector with `group_data` to obtain one intercept per row.

Shared and group intercepts

The regression module's `BernoulliLogitGLM` takes one shared intercept. Here `a[group_data]` supplies a separate intercept for each group. The dotted Bernoulli law uses the probability for each row.

Jessica Cox · bcox@mpp.mpg.de