Categories without enough observations? Merge levels at inference time with Bayes!

or “Effect fusion priors: a principled Bayesian approach for merging levels of a categorical variable”

Luna Fazio[1], Javier Enrique Aguilar Romero[1], Katja Ickstadt[1], Paul-Christian Bürkner[1]

[1] Department of Statistics, Technical University Dortmund

You might care about this if...

... your data looks like **this**

- **Categorical variable**
- **Levels with low counts**
- **Similar effects plausible**

301. ¿CUÁL ES EL ÚLTIMO AÑO O GRADO DE ESTUDIOS Y NIVEL QUE APROBÓ?

	Año	Grado	Centro de Estudios: Estatal	Centro de Estudios: No Estatal	
Sin nivel........................ 1					PASE A 302
Educación inicial 2			1	2	PASE A 302
Primaria incompleta.......... 3			1	2	PASE A 302
Primaria completa........... 4			1	2	PASE A 303
Secund. incompleta....... 5			1	2	PASE A 303
Secund. completa 6			1	2	
Básica especial 12			1	2	PASE A 302
Sup. no universitaria incompleta.....................7			1	2	
Sup. no universitaria completa........................8			1	2	
Sup. universitaria incompleta.....................9			1	2	
Sup. universitaria completa 10			1	2	
Maestría/Doctorado..... 11			1	2	

Source: questionnaire for Peru's *Encuesta Nacional de Hogares 2022*

What is the big idea?

- **Merging = implicit prior on differences.** When we merge levels, we are <u>fixing</u> the difference of the effects to zero.
- **Unsure about merging? Mixture model.** Pauger and Wagner (2019) proposed using a spike-and-slab prior on the differences between the coefficients:

$$(\beta_k - \beta_j) \mid \tau^2, \delta_{kj} \sim \mathrm{N}\left(0,\ \tau^2\left[\delta_{kj} + (1-\delta_{kj})r^{-1}\right]\right)$$

- **Equivalent to a prior on the coefficients.** Think good old multivariate normal, with a spicy covariance matrix.

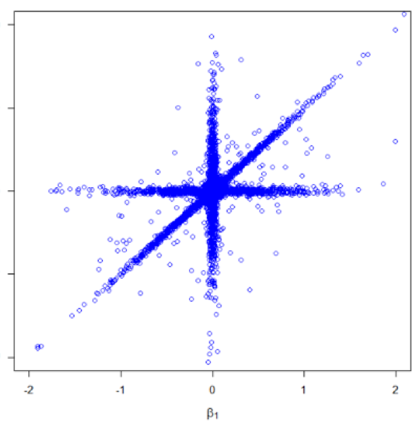


Fig. 1, Pauger and Wagner (2019)

Step-by-step: 3-level variable

Step 1: Construct the structure matrix

your variable:
choice A, choice B, choice C

$$\underbrace{(0,\ \beta_1,\ \beta_2)}_{\boldsymbol{\beta}} \leftrightarrow \begin{matrix} \beta_1 - 0 \overset{?}{=} 0 \\ \beta_2 - 0 \overset{?}{=} 0 \\ \beta_2 - \beta_1 \overset{?}{=} 0 \end{matrix} \Rightarrow \boldsymbol{\beta} \mid \tau^2, \mathbf{Q} \sim \mathrm{N}\left(\mathbf{0},\ \tau^2 \underbrace{\mathbf{Q}}_{\begin{bmatrix} \kappa_{10}+\kappa_{12} & -\kappa_{12} \\ -\kappa_{12} & \kappa_{20}+\kappa_{12} \end{bmatrix}}{}^{-1}\right)$$

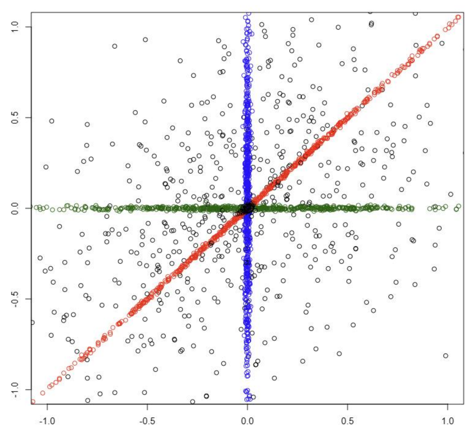
Step 2: Give the elements a bimodal distribution

$$\kappa_{kj} = \delta_{kj} + (1-\delta_{kj})r, \qquad \delta_{kj} \sim \mathrm{Bernoulli(p)}$$

some large constant

Step 3: Enjoy!

$$\begin{bmatrix} \kappa_{10}+\kappa_{12} & -\kappa_{12} \\ -\kappa_{12} & \kappa_{20}+\kappa_{12} \end{bmatrix} \Rightarrow$$



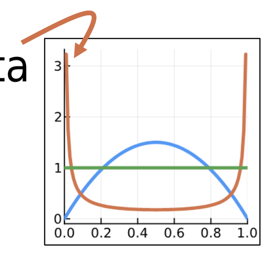
$$\leftrightarrow \begin{matrix} \beta_1 - 0 \overset{?}{=} 0 \\ \beta_2 - 0 \overset{?}{=} 0 \\ \beta_2 - \beta_1 \overset{?}{=} 0 \end{matrix}$$

So what is missing?

Generalizations and further analysis

- Using a discrete distribution limits the samplers that can be used for this prior.
- A "bathtub" beta* appears to be a suitable continuous replacement but we want to explore other options.
- We want to better understand the behavior with varying number of levels and strength of the signal.
- More interpretable parametrizations, usage guidelines, sensible defaults.

* "bathtub" beta



Implementation

- Pauger and Wagner provided an **R** library that implements a Gibbs sampler but it is no longer compatible with more recent versions of **R**.
- We already have a working prototype for **Julia** but the implementation and interface need to be polished.
- We would also like to provide an implementation in **Stan** and integrate it with **brms** for ease of use.
- <u>Anything else you would like to see?</u>

Sneak peek

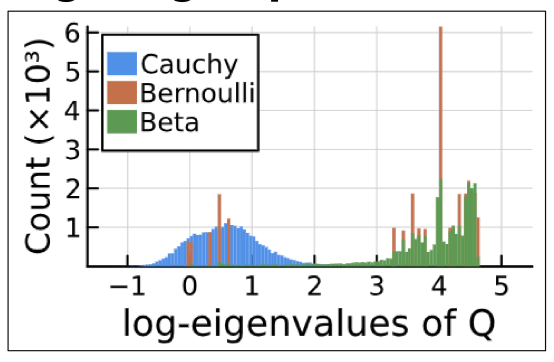
Turing.jl code for the 3-level case

```
@model function fxfus(d)
    τ ~ InverseGamma(5,2) # following P&W2019
    δ ~ filldist(d, 3) # user-specified!
    Q = Hermitian([
       sum(δ[[1,2]]) -δ[2]
       -δ[2] sum(δ[[2,3]]))
      ])
     # Coefficients
    β ~ MvNormal(zeros(2), τ*inv(Q))
    return (;Q,β)
end
```

Interface can be improved, but already usable!

```
@model function simpreg(X, y, d)
    β = @submodel fxfus(d)
    y ~ MvNormal(X * [0.0;β], σ*I)
end
```

Investigating impact of distribution



For more information

- Daniela Pauger, Helga Wagner. "Bayesian Effect Fusion for Categorical Predictors." Bayesian Analysis, June 2019.

- Also check our project's website

Sounds useful to you? Let's talk!

- **We are looking for case studies and potential applications.**

You can reach me at **bmfaziol@gmail.com**

= **lunafazio.github.io/effect-fusion**
(download this poster, code examples, future updates, etc.)

technische universität dortmund

fakultät statistik