Modèles hiérarchiques bayésiens 2

Contents

Données

Figure 1: Svalbard reindeer

Les données proviennent de mon post-doctorat. Nous analyserons la probabilité d'un renne d'avoir un bébé durant l'été. Dans ce système, un des plus importants facteurs environnementaux est la présence d'épisodes de pluie-sur-neige. Ceux-ci surviennent quand des précipitations surviennent au cours de l'hivers. Celles-ci gèlent et forment ensuite d'épaisse couche de glace bloquant l'accès aux ressources alimentaires.

Cependant, avec le réchauffement continue de l'arctique, certain chercheur croient que le ros n'aura plus d'effet. Quand il y a des épisodes de pluies durables suivi d'une période chaude, la pluie cause un dégagement des ressources alimentaires et a le temps de ruisseler avant de geler. Nous tenterons d'explorer ces changements dans l'effet du ros.

```
library(dplyr)
library(readr)
library(ggplot2)
library(tidyr)
```

```r
library(cowplot)
library(lubridate)

dat <- read_csv("../donnees/SvalbardDat.csv")
dat <- dat %>% mutate(age=year-yrbirth) %>% filter(age>1)

ros <- read_csv("../donnees/ROS.csv")
```

Nous transformons d'abord les prédicteurs:

- *rosNorm* est le logarithme de *ros*, normalisé pour avoir une moyenne de 0 et un écart-type de 1.
- *ages* est la normalisation de age pour avoir une moyenne de 0 et un écart-type de 1
- Nous créons une variable *age2* pour l'effet quadratique de l'âge.
- et finalement une variable *période* qui sépare l'étude en 5 périodes

1. Modèle bayésien de la probabilité de reproduction selon l'âge et du ros

Bien entendu, il faudra contrôler pour l'âge des individus. Le modèle sera donc un modèle binomial avec comme effet fixe, l'âge et son carré, et le ros. Les effets aléatoires consisteront de l'année, la période et une pente du ros variant selon la période.

Notes:

- La formule du modèle dans `brm` suit la même syntaxe que `lmer` pour la spécification des effets fixes et aléatoires.
- Bien qu'il serait possible d'ajouter l'interaction `age:ros`, l'année, l'ID, la densité et plusieurs autres variables contrôle. Nous les omettons ici afin de réduire le temps de calcul des modèles.

A Choisissez des distributions *a priori* pour les paramètres du modèle décrit ci-dessus. Voici un exemple de code où il ne manque que la spécification des distributions. Les quatre premières lignes définissent les distributions *a priori* pour l'ordonnée à l'origine et les coefficients des trois effets fixes, les trois suivantes définissent les distributions pour les écarts-types des effets aléatoires (`class = "sd"`), tandis que la dernière réfère à l'écart-type des observations individuelles (`class = "sigma"`).

```r
library(brms)
my_prior <- c(set_prior("", class = "Intercept"),
              set_prior("", class = "b", coef = "ages"),
              set_prior("", class = "b", coef = "age2"),
              set_prior("", class = "b", coef = "rosNorm"),
              set_prior("", class = "sd", coef = "Intercept", group = "id"),
              set_prior("", class = "sd", coef = "Intercept", group = "period"),
              set_prior("", class = "sd", coef = "rosNorm", group = "period"))
```

Il est recommandé de choisir des distributions normales dans tous les cas. Pour "sd", ces distributions seront interprétées comme des demi-normales car il est sous-entendu que ces paramètres sont ≥ 0. Pour choisir la moyenne et l'écart-type de chaque distribution normale, considérez l'interprétation de chaque paramètre et en particulier les échelles des prédicteurs `ros` , `ages` et `age2`. Dans bmrs, la famille utilisé sera `family=bernoulli("logit")`.

Quant aux écarts-types des effets aléatoires ("sd"), leur distribution *a priori* peut avoir la même largeur que celle du coefficient "b" correspondant.

B Tirez maintenant un échantillon de la distribution conjointe *a priori* des paramètres avec `brm`. Je suggère de spécifier `chains = 1, iter = 1500, warmup = 1000` pour produire une seule chaîne de Markov avec 1000 itérations de rodage et 500 itérations d'échantillonnage. Visualisez ensuite la distribution de `calf` prédite pour chaque itération des paramètres *a priori*.

En raison du grand nombre d'effets estimés et du fait que nous n'imposons que des contraintes légères sur chaque distribution *a priori*, on doit s'attendre à des valeurs extrêmes voire impossibles (grandes valeurs positives et négatives); l'important est que la densité soit plus grande dans une plage de valeurs réalistes. Il peut être utile de faire un "zoom" sur une partie du graphique `ggplot` en y ajoutant `coord_cartesian(xlim = c(..., ...), ylim = c(..., ...))` avec des limites en x et y.

C Ajustez maintenant le modèle avec `brm`. Vous pouvez réduire le nombre de chaînes de Markov à 2 pour sauver du temps, mais conservez les valeurs par défaut pour le nombre d'itérations. (Vous pouvez ignorer l'avertissement selon lequel la taille effective de l'échantillon ou ESS est faible.) Comment pouvez-vous évaluer la convergence du modèle?

D Comparez la magnitude du coefficient de `rosNorm` à celle des effets aléatoires. Qu'est-ce que cette comparaison vous apprend?

E Vérifier les prédictions a posteriori: Appliquez `predict` au modèle pour obtenir la moyenne, l'écart-type et l'intervalle à 95% pour la prédiction *a posteriori*. Vous pouvez donner un data.frame à l'argument newdata pour obtenir les prédictions voulues (`expand.grid(rosNorm=seq(-1.6,1.6,l=30),period=unique(dat$period),...)`). Illustrez les prédictions du modèle et leurs intervalles de crédibilité pour les différentes périodes pour un individu de 7 ans.