Practical 3 - Exercises in `rstanarm`

Andrew Parnell

Introduction

Welcome to Practical 3, some exercises in `rstanarm`. In this practical we will

- Fit some basic linear mixed models and generalised linear mixed models
- Explore the output
- Create some plots using `bayesplot`

This practical sticks to the same format/questions as practical 2 but uses `rstanarm` instead for producing the results.

Exercise 1

Go back and make sure you're happy with all the code from today's lectures.

Linear mixed effects models

Let's start by fitting a linear mixed model to a new data set. We're going to use the `prostate` data set in the `data` folder. The response variable is going to be `lpsa` the log of the prostate specific antigen value. All the others variables in the data set are the covariates but we are going to focus specifically on the continuous covariate `lcavol` (the log of the cancer volume) and the discrete covariate `gleason` which gives the Gleason grade (a measure of how severe the cancer is).

Exercise 2

1. Copy your code in from yesterday to load in the data and use suitable plotting commands to look at the relationship between the response and the covariate, possibly also varying by the discrete .
2. Fit a fixed effects model to the data using `stan_lm`, first with just `lcavol` and then with an interaction term between `lcavol` and `gleason`. Try to interpret the output (hint: make sure `gleason` is a factor).
3. Fit a set of mixed effects models with varying intercepts and/or slopes for the model above using the `stan_lmer` function. Create some plots to verify the fits and check the varying nature of the random effects. Compare the random effects to the fixed effect values you got in the previous step.
4. Create a plot of the residuals (y-axis) vs fitted values (x-axis). The answer on how to do this is in the answer script but see if you can find it yourself.
5. Ensure convergence from the $\hat{R}$ values. Compute a posterior predictive check of the models using `pp_check` and determine which you think fits the data best.

This data set also contains a column called `train` which splits the data into a training set and a test set. We would like to fit your chosen model to the training set and see how it performs on the test set.

Exercise 3

1. Subset the data so you are left with just the rows where `train == T`. Fit your best mixed effects model to that data set and check performance

2. Use the `posterior_predict` function to get predicted values of `lpsa` for the training set data. Check that the predictions agree with the true values (via e.g. a plot or a correlation score). (Hint: `posterior_predict` will give you a full set of posterior samples which you will need to summarise using apply)
3. Now use the `posterior_predict` function to get predictions for the data you removed (i.e. `train == F`). See if you can produce predictions that remove the effect of the random effects (hint: see the `posterior_predict` help file). Do the random effects improve the test set predictions?

A generalised linear mixed model example

Let's move on to a glmm example. We're going to use the `pollen` data set, which is a set of pollen counts which vary by two climate markers. We're going to use the response variable `Betula` (Birch). The two covariates (both continuous) are Mean Temperature of the coldest month (MTCO) and Growing Degree Days above 5 (GDD5; also known as the annual temperature sum above 5 degrees).

Exercise 4

1. Load in the data and standardise the two climate variables using `scale`. create a plot of the count against each of the continuous covariates. Also see if you can plot the counts against both variables simultaneously (harder)
2. Try and fit some fixed effects glms to the data to get an idea of the relationships. Make sure to check the diagnostics

---

To fit some glmms, we're going to partition the `MTCO` variable into 4 levels. We're then going to fit some Poisson and negative binomial models to see which works best. Be aware that some of the relationships are non-linear and getting models which fit the data well is challenging!

Exercise 5

1. From yesterday, create a new variable `MTCO_cut` which is defined as: `cold_winter` if `MTCO` $\leq$ 17, `mild_winter` if -17 $<$ `MTCO` $\leq$ -8, `warm_winter` if -8 $<$ `MTCO` $\leq$ 0, and `hot_winter` if `MTCO` $>$ 0. Hint: use the `cut` function. Create a table of `MTCO_cut` values and see if 1. Fit some initial Poisson glmms, perhaps using the structure you might have learnt from the previous exercise (i.e. perhaps a non-linear relationship?). Check the fit of these models
2. Fit some Negative Binomial glmms. Note that to do this you have to use the `stan_glmer.nb` function which means you don't need a `family` command but otherwise all is the same. Does this improve the fit? Use the tools we have learnt to help decide which models are best. ***

Others exercises

1. Tomorow we will be using `rstan` to fit some of these models instead. See if you can translate some of the simpler models into `rstan` format. Note this will involve reading ahead a bit in the notes.
2. Have a first go at running `rstanarm` on your chosen data set. Try and fit the simplest possible model you can think of first, and slowly make it more complicated. Remember to start with a plot of your data and make sure you keep plotting/tabulating your results to check that it makes sense.