Bayesian Inference Assignment 2

Due Monday May 6 by 8:45 AM

For this homework, we are going to analyze data from the General Social Survey, which is a survey on a wide variety of topics asked to a sample of adults in the United States every two years. Download the file called `GSS.rds` from Classes to the same directory as your RMarkdown file, which you can then load into R with

```
GSS <- readRDS("GSS.rds") # actually a subset that is not missing on the outcome
```

In particular, we are going to model responses to a question on marriage equality that was first asked in 1988, not asked again until 2004, but asked in every wave thereafter. The text of the question remained the same in all years that it was asked and reads as

> "Do you agree or disagree [that] homosexual couples should have the right to marry one another?"

As can be seen from

```
levels(GSS$marhomo)
```

```
## [1] "strongly agree"             "agree"
## [3] "neither agree nor disagree" "disagree"
## [5] "strongly disagree"
```

there are five (valid) response categories that can be ordered by strength of opposition.

As background, in the 1990s essentially all states had laws defining marriage to be only between a man and a woman. But about ten years ago some states started to acknowledge "civil unions", then a few permitted same-sex marriages via legislation, after which many state and circuit courts struck down prohibitions on same-sex marriage, and finally the Supreme Court in 2015 did so for the United States as a whole. The topic has always been polarizing, but in a relatively short amount of time (for a social issue) public opinion shifted from a strong majority opposing same-sex marriages to a strong majority supporting them.

There are thousands of variables you could use to predict with (although the geographic variables for State, Primary sampling unit, County, and Census track are not included in the *public* version of the GSS). You could read through the http://gss.norc.org/get-documentation, especially the "Index to Data Set" chapter, which has a concise listing of variable descriptions. Or it can be useful to search for variables using the online https://gssdataexplorer.norc.org/variables/vfilter. Finally, I have output the data dictionary as GSS_dictionary.html on Classes, which has the variable descriptions along with the definitions of the response categories. You can get the answer categories for factor variables in R by doing

```
vals <- sapply(GSS, FUN = attr, which = "levels")
```

1 Prior Predictive Distribution without Predictors

Using the **brms** package, you can call `brm` as usual but with the non-default arguments `family = cumulative, sample_prior = "only"` to not condition on the observed data and thereby draw from the prior distribution of the unknowns in an ordered logit model. You can then evaluate the "posterior" distribution of the linear predictor by calling `posterior_linpred` on the object returned by `brm`, which is really the prior distribution of the linear predictor if the data were not conditioned on originally. The `posterior_linpred` function in this case produces a $S \times N \times 4$ array whose elements contain the difference between η_n and the j-th cutpoint on the s-th simulation. If you call `posterior_linpred` but specify the non-default argument `transform = TRUE`, the resulting array is $S \times N \times 5$ and contains the predicted probability that the n-th observation would fall in each of the five response categories on the s-th simulation. You could also call `posterior_predict` on the object produced by `brm` to obtain a $S \times N$ matrix whose

elements are integers, indicating which of the five answer categories the n-th person was predicted to answer on the s-th simulation. Any of these functions can be used to judge whether your priors are reasonable.

When calling `brm`, you should specify the `prior` argument to be something other than the default priors. Start with the simplest model that has no predictors and assumes public opinion is constant over time — so the formula is just `marhomo ~ 1` and adjust you priors on the cutpoints to obtain plausible distribution of public opinion (that you can perhaps think of as "average" opinion over this time period). As you can see from `get_prior`, the cutpoints are given marginal priors from some distribution (by default `student_t`) and then get truncated in the Stan program to enforce the ordering constraint.

2 Prior Predictive Distribution with Time

Next, add a time dimension to your model. There is a (numeric) `year` variable in `GSS`, although it is not necessarily the case that public opinion changed linearly over this time period. Again draw from the prior distribution of the unknowns in this slightly more complex model and tweak your priors so that it implies a plausible shift over time in favor of same-sex marriage.

3 Prior Predictive Distribution with Time and Other Predictors

Finally, add predictors to your model that vary across individuals that attempt to explain the bulk of the variation in public opinion on same-sex marriage. Make sure that the priors still imply a reasonable distribution of aggregate public opinion over time.

4 Posterior Distribution

Now call the `brm` function with the same arguments as in the previous question, except with the (default) `sample_prior = "no"` argument in order to condition on the data and draw from the posterior distribution.

Use the `plot` method for the result of the `marginal_effects` function called with the (non-default) `ordinal = TRUE` argument to produce a useful plot showing how the posterior probability of falling in each of the answer categories varies as a function of some predictor.

5 Model Comparison by ELPD

Estimate an alternative model to the one in the previous question, and then use the `loo` and `loo_compare` functions in the **loo** package to compare these two models. Which model is expected to best predict the 2020 GSS data on this question and in qualitative terms, is the difference between models substantial or would you be essentially indifferent between them?

6 Model Comparison by Posterior Probability

If you specify the (non-defaut) `save_all_pars = TRUE` argument to `brm` when estimating the previous two models, you can also use the functions in the **bridgesampling** package to estimate the posterior probability that each model is correct under the strong assumption that one of the two models is correct. Call `bridge_sampler` and `error_measures` on each of the two candidate models and then call `post_prob` on the two results from `bridge_sampler`. How do these posterior probabilities over models differ numerically and conceptually from the result of calling `loo_model_weights` on the two models?