8.5. Practice

Problems are labeled Easy (**E**), Medium (**M**), and Hard (**H**).

8E1. For each of the causal relationships below, name a hypothetical third variable that would lead to an interaction effect.

(1) Bread dough rises because of yeast.
(2) Education leads to higher income.
(3) Gasoline makes a car go.

8E2. Which of the following explanations invokes an interaction?

(1) Caramelizing onions requires cooking over low heat and making sure the onions do not dry out.
(2) A car will go faster when it has more cylinders or when it has a better fuel injector.
(3) Most people acquire their political beliefs from their parents, unless they get them instead from their friends.
(4) Intelligent animal species tend to be either highly social or have manipulative appendages (hands, tentacles, etc.).

8E3. For each of the explanations in **8E2**, write a linear model that expresses the stated relationship.

8M1. Recall the tulips example from the chapter. Suppose another set of treatments adjusted the temperature in the greenhouse over two levels: cold and hot. The data in the chapter were collected at the cold temperature. You find none of the plants grown under the hot temperature developed any blooms at all, regardless of the water and shade levels. Can you explain this result in terms of interactions between water, shade, and temperature?

8M2. Can you invent a regression equation that would make the bloom size zero, whenever the temperature is hot?

8M3. In parts of North America, ravens depend upon wolves for their food. This is because ravens are carnivorous but cannot usually kill or open carcasses of prey. Wolves however can and do kill and tear open animals, and they tolerate ravens co-feeding at their kills. This species relationship is generally described as a "species interaction." Can you invent a hypothetical set of data on raven population size in which this relationship would manifest as a statistical interaction? Do you think the biological interaction could be linear? Why or why not?

8M4. Repeat the tulips analysis, but this time use priors that constrain the effect of water to be positive and the effect of shade to be negative. Use prior predictive simulation. What do these prior assumptions mean for the interaction prior, if anything?

8H1. Return to the `data(tulips)` example in the chapter. Now include the `bed` variable as a predictor in the interaction model. Don't interact `bed` with the other predictors; just include it as a main effect. Note that `bed` is categorical. So to use it properly, you will need to either construct dummy variables or rather an index variable, as explained in Chapter 5.

8H2. Use WAIC to compare the model from **8H1** to a model that omits `bed`. What do you infer from this comparison? Can you reconcile the WAIC results with the posterior distribution of the `bed` coefficients?

8H3. Consider again the `data(rugged)` data on economic development and terrain ruggedness, examined in this chapter. One of the African countries in that example, Seychelles, is far outside the cloud of other nations, being a rare country with both relatively high GDP and high ruggedness. Seychelles is also unusual, in that it is a group of islands far from the coast of mainland Africa, and its main economic activity is tourism.

(a) Focus on model `m8.5` from the chapter. Use WAIC pointwise penalties and PSIS Pareto k values to measure relative influence of each country. By these criteria, is Seychelles influencing the results? Are there other nations that are relatively influential? If so, can you explain why?

(b) Now use robust regression, as described in the previous chapter. Modify `m8.5` to use a Student-t distribution with $\nu = 2$. Does this change the results in a substantial way?

8H4. The values in `data(nettle)` are data on language diversity in 74 nations.[143] The meaning of each column is given below.

(1) `country`: Name of the country
(2) `num.lang`: Number of recognized languages spoken
(3) `area`: Area in square kilometers
(4) `k.pop`: Population, in thousands
(5) `num.stations`: Number of weather stations that provided data for the next two columns
(6) `mean.growing.season`: Average length of growing season, in months
(7) `sd.growing.season`: Standard deviation of length of growing season, in months

Use these data to evaluate the hypothesis that language diversity is partly a product of food security. The notion is that, in productive ecologies, people don't need large social networks to buffer them against risk of food shortfalls. This means cultural groups can be smaller and more self-sufficient, leading to more languages per capita. Use the number of languages per capita as the outcome:

R code 8.27
```
d$lang.per.cap <- d$num.lang / d$k.pop
```

Use the logarithm of this new variable as your regression outcome. (A count model would be better here, but you'll learn those later, in Chapter 11.) This problem is open ended, allowing you to decide how you address the hypotheses and the uncertain advice the modeling provides. If you think you need to use WAIC anyplace, please do. If you think you need certain priors, argue for them. If you think you need to plot predictions in a certain way, please do. Just try to honestly evaluate the main effects of both `mean.growing.season` and `sd.growing.season`, as well as their two-way interaction. Here are three parts to help. (a) Evaluate the hypothesis that language diversity, as measured by `log(lang.per.cap)`, is positively associated with the average length of the growing season, `mean.growing.season`. Consider `log(area)` in your regression(s) as a covariate (not an interaction). Interpret your results. (b) Now evaluate the hypothesis that language diversity is negatively associated with the standard deviation of length of growing season, `sd.growing.season`. This hypothesis follows from uncertainty in harvest favoring social insurance through larger social networks and therefore fewer languages. Again, consider `log(area)` as a covariate (not an interaction). Interpret your results. (c) Finally, evaluate the hypothesis that `mean.growing.season` and `sd.growing.season` interact to synergistically reduce language diversity. The idea is that, in nations with longer average growing seasons, high variance makes storage and redistribution even more important than it would be otherwise. That way, people can cooperate to preserve and protect windfalls to be used during the droughts.

8H5. Consider the `data(Wines2012)` data table. These data are expert ratings of 20 different French and American wines by 9 different French and American judges. Your goal is to model `score`, the subjective rating assigned by each judge to each wine. I recommend standardizing it. In this problem, consider only variation among judges and wines. Construct index variables of `judge` and `wine` and then use these index variables to construct a linear regression model. Justify your priors. You should end up with 9 judge parameters and 20 wine parameters. How do you interpret the variation among individual judges and individual wines? Do you notice any patterns, just by plotting the differences? Which judges gave the highest/lowest ratings? Which wines were rated worst/best on average?

8H6. Now consider three features of the wines and judges:

(1) `flight`: Whether the wine is red or white.
(2) `wine.amer`: Indicator variable for American wines.
(3) `judge.amer`: Indicator variable for American judges.

Use indicator or index variables to model the influence of these features on the scores. Omit the individual judge and wine index variables from Problem 1. Do not include interaction effects yet. Again justify your priors. What do you conclude about the differences among the wines and judges? Try to relate the results to the inferences in the previous problem.

8H7. Now consider two-way interactions among the three features. You should end up with three different interaction terms in your model. These will be easier to build, if you use indicator variables. Again justify your priors. Explain what each interaction means. Be sure to interpret the model's predictions on the outcome scale (`mu`, the expected score), not on the scale of individual parameters. You can use `link` to help with this, or just use your knowledge of the linear model instead. What do you conclude about the features and the scores? Can you relate the results of your model(s) to the individual judge and wine inferences from **8H5**?

9.6. Summary

This chapter has been an informal introduction to Markov chain Monte Carlo (MCMC) estimation. The goal has been to introduce the purpose and approach MCMC algorithms. The major algorithms introduced were the Metropolis, Gibbs sampling, and Hamiltonian Monte Carlo algorithms. Each has its advantages and disadvantages. The `ulam` function in the `rethinking` package was introduced. It uses the Stan (mc-stan.org) Hamiltonian Monte Carlo engine to fit models as they are defined in this book. General advice about diagnosing poor MCMC fits was introduced by the use of a couple of pathological examples. In the next chapters, we use this new power to learn new kinds of models.

9.7. Practice

Problems are labeled Easy (**E**), Medium (**M**), and Hard (**H**).

9E1. Which of the following is a requirement of the simple Metropolis algorithm?

(1) The parameters must be discrete.
(2) The likelihood function must be Gaussian.
(3) The proposal distribution must be symmetric.

9E2. Gibbs sampling is more efficient than the Metropolis algorithm. How does it achieve this extra efficiency? Are there any limitations to the Gibbs sampling strategy?

9E3. Which sort of parameters can Hamiltonian Monte Carlo not handle? Can you explain why?

9E4. Explain the difference between the effective number of samples, `n_eff` as calculated by Stan, and the actual number of samples.

9E5. Which value should `Rhat` approach, when a chain is sampling the posterior distribution correctly?

9E6. Sketch a good trace plot for a Markov chain, one that is effectively sampling from the posterior distribution. What is good about its shape? Then sketch a trace plot for a malfunctioning Markov chain. What about its shape indicates malfunction?

9E7. Repeat the problem above, but now for a trace rank plot.

9M1. Re-estimate the terrain ruggedness model from the chapter, but now using a uniform prior for the standard deviation, `sigma`. The uniform prior should be `dunif(0,1)`. Use `ulam` to estimate the posterior. Does the different prior have any detectible influence on the posterior distribution of `sigma`? Why or why not?

9M2. Modify the terrain ruggedness model again. This time, change the prior for `b[cid]` to `dexp(0.3)`. What does this do to the posterior distribution? Can you explain it?

9M3. Re-estimate one of the Stan models from the chapter, but at different numbers of `warmup` iterations. Be sure to use the same number of sampling iterations in each case. Compare the `n_eff` values. How much warmup is enough?

9H1. Run the model below and then inspect the posterior distribution and explain what it is accomplishing.

R code 9.28

```
mp <- ulam(
    alist(
        a ~ dnorm(0,1),
        b ~ dcauchy(0,1)
    ), data=list(y=1) , chains=1 )
```

Compare the samples for the parameters `a` and `b`. Can you explain the different trace plots? If you are unfamiliar with the Cauchy distribution, you should look it up. The key feature to attend to is that it has no expected value. Can you connect this fact to the trace plot?

9H2. Recall the divorce rate example from Chapter 5. Repeat that analysis, using `ulam` this time, fitting models `m5.1`, `m5.2`, and `m5.3`. Use `compare` to compare the models on the basis of WAIC or PSIS. To use WAIC or PSIS with `ulam`, you need add the argument `log_log=TRUE`. Explain the model comparison results.

9H3. Sometimes changing a prior for one parameter has unanticipated effects on other parameters. This is because when a parameter is highly correlated with another parameter in the posterior, the prior influences both parameters. Here's an example to work and think through.

Go back to the leg length example in Chapter 6 and use the code there to simulate height and leg lengths for 100 imagined individuals. Below is the model you fit before, resulting in a highly correlated posterior for the two beta parameters. This time, fit the model using `ulam`:

R code 9.29

```
m5.8s <- ulam(
    alist(
        height ~ dnorm( mu , sigma ) ,
        mu <- a + bl*leg_left + br*leg_right ,
        a ~ dnorm( 10 , 100 ) ,
        bl ~ dnorm( 2 , 10 ) ,
        br ~ dnorm( 2 , 10 ) ,
        sigma ~ dexp( 1 )
    ) , data=d, chains=4,
    start=list(a=10,bl=0,br=0.1,sigma=1) )
```

Compare the posterior distribution produced by the code above to the posterior distribution produced when you change the prior for `br` so that it is strictly positive:

R code 9.30

```
m5.8s2 <- ulam(
    alist(
        height ~ dnorm( mu , sigma ) ,
        mu <- a + bl*leg_left + br*leg_right ,
        a ~ dnorm( 10 , 100 ) ,
        bl ~ dnorm( 2 , 10 ) ,
        br ~ dnorm( 2 , 10 ) ,
        sigma ~ dexp( 1 )
    ) , data=d, chains=4,
    constraints=list(br="lower=0"),
    start=list(a=10,bl=0,br=0.1,sigma=1) )
```

Note the `constraints` list. What this does is constrain the prior distribution of `br` so that it has positive probability only above zero. In other words, that prior ensures that the posterior distribution for `br` will have no probability mass below zero. Compare the two posterior distributions for `m5.8s` and `m5.8s2`. What has changed in the posterior distribution of both beta parameters? Can you explain the change induced by the change in prior?

9H4. For the two models fit in the previous problem, use WAIC or PSIS to compare the effective numbers of parameters for each model. You will need to use `log_lik=TRUE` to instruct `ulam` to compute the terms that both WAIC and PSIS need. Which model has more effective parameters? Why?

9H5. Modify the Metropolis algorithm code from the chapter to handle the case that the island populations have a different distribution than the island labels. This means the island's number will not be the same as its population.

9H6. Modify the Metropolis algorithm code from the chapter to write your own simple MCMC estimator for globe tossing data and model from Chapter 2.

9H7. Can you write your own Hamiltonian Monte Carlo algorithm for the globe tossing data, using the R code in the chapter? You will have to write your own functions for the likelihood and gradient, but you can use the `HMC2` function.