Statistical models

2022-09-27

Contents

Homework

Homework Exercises:

Self-study: Work these exercises, but do not turn them in.

- Exercises 6E1, 6E3, 6E4
- Exercises 7E1, 7E3, 7E4

Turn in: Work these exercises and turn them in.

- Exercises 6M2–6M3
- Exercises 7M2–7M6

Optional: The following exercises are optional. You can turn them in for extra credit.

- Exercise 6H1

Notes on Homework:

Exercise 7M3:

For exercise 7M3, I recommend using the hominin model 7.4 in section 7.1:

First, create the data (code 7.1 and 7.2), and then use model 7.4 from code 7.8, on p. 198. Make several variations, in which you change the standard deviation on the prior for `b` from 10 to smaller values, such as 5, 2, 1, 0.5, 0.1, and 0.01.

```
lst_7.1 <- alist(
brain_std ~ dnorm(mu, exp(log_sigma)),
mu <- a + b[1] * mass_std + b[2] * mass_std^2 +
      b[3] * mass_std^3 + b[4] * mass_std^4,
a ~ dnorm(0.5, 1),
b ~ dnorm(0, 10),
log_sigma ~ dnorm(0, 1)
)

lst_7.1a <- alist(
brain_std ~ dnorm(mu, exp(log_sigma)),
mu <- a + b[1] * mass_std + b[2] * mass_std^2 +
      b[3] * mass_std^3 + b[4] * mass_std^4,
a ~ dnorm(0.5, 1),
```

```
b ~ dnorm(0, 5),
log_sigma ~ dnorm(0, 1)
)
```

and so forth, for smaller and smaller values of the standard deviation on the prior for `b`.

Then fit each model to the data like this:

```
mdl_7.1 <- quap(lst_7.1, data = d, start = list(b = rep(0,4)))
mdl_7.1a <- quap(lst_7.1a, data = d, start = list(b = rep(0,4)))
...
```

Use `precis` to inspect the values of the parameters and see how they change as you apply more and more informative priors

Then you can sample from the models, following the code in 7.10

```
new_data <- data.frame(mass_std = seq(min(d$mass_std), max(d$mass_std),
                                      length.out = 100))
lnk_7.1 <- link(mdl_7.1, data = new_data)
lnk_7.1a <- link(mdl_7.1a, data = new_data)
...
```

Finally, you can plot the models

```
mu_7.1 <- apply(lnk_7.1, 2, mean)
ci_7.1 <- apply(lnk_7.1, 2, PI)

mu_7.1a <- apply(lnk_7.1a, 2, mean)
ci_7.1a <- apply(lnk_7.1a, 2, PI)
...

plot(brain_std ~ mass_std, data = d)
lines(new_data$mass_std, mu_7.1)
shade(ci_7.1, new_data$mass_std)

plot(brain_std ~ mass_std, data = d)
lines(new_data$mass_std, mu_7.1a)
shade(ci_7.1a, new_data$mass_std)
...
```

and so forth.

As you make the prior for `b` narrower and narrower, how do things change in terms of overfitting and underfitting?

Exercise 7M4:

For exercise 7M4, I recommend using the fungus treatment example from section 6.2 or the educational attainment example from section 6.3.2.

Start with the code for creating the data (code 6.13 for the fungus example, or 6.25–6.26 for the education example). Modify the code to generate a data-frame with 200 rows ($N = 200$)

Then instead of using the code for fitting models from 6.15–6.17 or 6.27–6.28, make the formula lists separately and then call `quap` with different amounts of data:

```
lst_6.6 <- alist(
  h1 ~ dnorm(mu, sigma),
  mu <- h0 * p,
  p ~ dlnorm(0, 0.25),
  sigma ~ dexp(1)
```

```
)
```

and make similar lists for models 6.7 and 6.8.

If you're using the educational attainment model, there are only two models in the book, so I'd recommend making up a 3rd model with fewer parameters (maybe only look at `G` and ignore `P`).

You can pick *n* rows from the data frame *d* using the `slice_sample` function:

```
d_20 <- slice_sample(d, n = 20)
d_50 <- slice_sample(d, n = 50)
d_100 <- slice_sample(d, n = 100)
```

Now you can fit each model to a different amount of data:

```
mdl_6.6_20 <- quap(lst_6.6, data = d_20)
```

First, for a single model, vary the amount of data and see how WAIC changes with the number of samples.

Then fit the different models to the same data (`d_20`, `d_50`, `d_100`), and use the `compare` function to see which has the smallest WAIC. Are the results consistent for the different amounts of data (20, 50, or 100 points)?

Next, compare models fit to different amounts of data: `mdl_6.6.20`, `mdl_6.7_50`, and `mdl_6.8_100` and other combinations. Does the `compare` function rank the models differently when you compare models fit to different amounts of data? (You will see a warning from `compare()` when you do this).