

Model Averaging

2020-02-04

Preparation

In this set of notes, we will learn about multimodel inference, an alternative to selecting a “best” model from a set of candidate models. To do so, we will use the *nels.csv* dataset (see the [data codebook](#)) to examine various prediction models to explain variation in students’ school achievement.

```
# Load libraries
library(broom)
library(educate) #Need version 0.1.0.1
library(MuMIn)
library(patchwork)
library(tidyverse)

# Import data
nels = read_csv("~/Documents/github/epsy-8252/data/nels.csv")

# View data
head(nels)
```

```
# A tibble: 6 x 6
  social10 self_esteem locus   ses   gpa parent_ed
    <dbl>      <dbl> <dbl> <dbl> <dbl>    <dbl>
1    61.7      -0.1  -0.14 -0.563  3.8      2
2    47.0     -0.45 -0.580  0.123  2.5      3
3    50.5       0.33 -0.59   0.229  2.8      3
4    56.5     -0.02  0.07   0.687  3.5      6
5    55.3     -0.09 -0.85   0.633  3.3      5
6    39.7     -0.28  0.07   0.992  2.5      5
```

Working Hypotheses and Candidate Models

Here, imagine a researcher who has five scientific working hypotheses about what underlies students’ achievement:

- **H1:** Student achievement is a function of SES, prior achievement, and parent education level.
- **H2:** Student achievement is a function of SES, prior achievement, and parent education level, but also of the student’s level of self-esteem.
- **H3:** Student achievement is a function of SES, prior achievement, and parent education level, but also of the student’s degree of locus-of-control.
- **H4:** Student achievement is a function of SES, prior achievement, and parent education level, but also of the student’s level of self-esteem and degree of locus-of-control.
- **H5:** Student achievement is a function of SES, prior achievement, and parent education level, but also of the interaction between student’s level of self-esteem and degree of locus-of-control.

You should be able to tell by the scientific hypotheses, that this scenario represents a very exploratory approach to examining the underlying structure of the models (i.e., which predictors seem important). This is fine, we just need to acknowledge this so that our results are not seen as “final”. Results from such an analysis are useful in helping plan a confirmatory analysis that can be carried out with new data.

The five working hypotheses are translated into the following five statistical models:

$$\text{Model 1 : Achievement}_i = \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \epsilon_i$$

$$\text{Model 2 : Achievement}_i = \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \beta_4(\text{Self-Esteem}_i) + \epsilon_i$$

$$\text{Model 3 : Achievement}_i = \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \beta_4(\text{Locus-of-Control}_i) + \epsilon_i$$

$$\text{Model 4 : Achievement}_i = \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \beta_4(\text{Self-Esteem}_i) + \beta_5(\text{Locus-of-Control}_i) + \epsilon_i$$

$$\text{Model 5 : Achievement}_i = \beta_0 + \beta_1(\text{SES}_i) + \beta_2(\text{Prior GPA}_i) + \beta_3(\text{Parent Education}_i) + \beta_4(\text{Self-Esteem}_i) + \beta_5(\text{Locus-of-Control}_i) + \beta_6(\text{Self-Esteem}_i)(\text{Locus-of-Control}_i) + \epsilon_i$$

These models are then fitted to the data.

```
# Fit Model 1
lm.1 = lm(social10 ~ 1 + ses + gpa + parent_ed, data = nels)

# Fit Model 2
lm.2 = lm(social10 ~ 1 + ses + gpa + parent_ed + self_esteem, data = nels)

# Fit Model 3
lm.3 = lm(social10 ~ 1 + ses + gpa + parent_ed + locus, data = nels)

# Fit Model 4
lm.4 = lm(social10 ~ 1 + ses + gpa + parent_ed + self_esteem + locus, data = nels)

# Fit Model 5
lm.5 = lm(social10 ~ 1 + ses + gpa + parent_ed + self_esteem + locus +
          self_esteem:locus, data = nels)
```

We can compute information-theoretic summaries (e.g., AICc, model probabilities) for these candidate models. To do this, we will use the `model.sel()` function from the **MuMIn** package.

```
#Create table
model.sel(list(lm.1, lm.2, lm.3, lm.4, lm.5))
```

Model selection table

	(Int)	gpa	prn_ed	ses	slf_est	lcs	lcs:slf_est	df	logLik	AICc	delta
3	34.42	5.064	0.4158	3.241		1.369		6	-2603.484	5219.1	0.00
4	34.46	5.050	0.4143	3.237	0.3487	1.165		7	-2603.284	5220.7	1.64
5	34.68	5.028	0.4171	3.223	0.3767	1.263	-0.7216	8	-2602.622	5221.4	2.36
2	34.15	5.169	0.4114	3.321	0.9476			6	-2605.244	5222.6	3.52
1	33.78	5.304	0.4153	3.393				5	-2607.343	5224.8	5.68

weight

3	0.505
4	0.223
5	0.156
2	0.087
1	0.029

Models ranked by AICc(x)

The model evidence is less clear about which model is supported. The model having the most empirical evidence (given the data and candidate set of models) is Model 3 (the model that includes the locus-of-control main effect along with the three covariates). This model has a probability (AICc weight) of 0.51. This level of model evidence is far from overwhelming. Models 4 (Main effects of self-esteem and locus-of-control), 5 (the interaction model), and 2 also have a fair amount of empirical support.

Anytime we adopt a model from a set of candidate models (whether through information-theoretic approaches or through statistical inference), we have introduced some amount of uncertainty via the model selection process; referred to as **model selection uncertainty**.

Think about having multiple replicate sets of data and fitting the same candidate models to each set of replication data. We would likely find that the “best” model varies across replicate sets of data. This is model selection uncertainty.

The amount of the model selection uncertainty depends on the level of empirical evidence that supports one model over all others. In our example here, we would introduce quite a lot of model selection uncertainty since the empirical evidence for Model 3 is not overwhelming; the evidence also supports Models 4, 5, and 2.

This is important for a couple reasons. First, adopting one model when there is uncertainty that it is the “best” model may be misleading for the broader scientific enterprise. For example, consider trying to understand the effect of self-esteem on achievement. If we adopt Model 3 there is no effect of self-esteem. However, if Model 4 is actually closest to reality there is a positive effect of self-esteem on achievement. And, if Model 5 is closest to reality, then the effect of self-esteem on achievement depends on locus-of-control. Which are we to believe?

Another reason to think about model selection uncertainty is that any inferential results we report for a model are based around sampling variation being the only source of uncertainty. When there is model selection uncertainty, this is not the case. Increasing uncertainty directly affects the size of the standard errors which in turn affect the magnitude of the *p*-values and confidence intervals.

Multimodel Inference

One solution to dealing with the problem of model selection uncertainty is to use *multimodel inference*. In this approach, information from all (or many) the models included in the candidate set is used to make statistical inferences. This approach to inference is rather new to the statistical sciences and the most commonly used method in multimodel inference is **model averaging**.

Model Averaging

In model averaging, we compute the coefficients and standard errors for the various predictors by incorporating information from these estimates from all the models in the candidate set. Rather than simple averaging, the estimates are weighted by the model probabilities. A simple averaging would give us a coefficient of

$$\hat{\beta}_{\text{Self-Esteem}} = \frac{0 + 0.3487 + 0.3767 + 0.9476 + 0}{5} = 0.3346$$

However, this assumes that we believe each of the models equally. This is not true. The model probabilities we computed in Table 1 suggest that we should give the most weight to the coefficients in Model 3, then Model 4, etc. We can incorporate this by computing a weighted average where the weights we use are the model probabilities from the table of model evidence.

$$\hat{\beta}_{\text{Self-Esteem}} = \frac{0(0.505) + 0.3487(0.223) + 0.3767(0.156) + 0.9476(0.087) + 0(0.029)}{1} = 0.218$$

Note that the denominator in the weighted average is simply the sum of the weights, which is 1. In general, we can express the estimate for the k th coefficient, $\hat{\beta}_k$, as

$$\hat{\beta}_k = w_j(\beta_{kj})$$

where w_j is the AICc weight (or model probability) for candidate model j , and β_{kj} is the estimated coefficient for predictor k in candidate model j .

Conceptually, the weighting across each of the models *shrinks* the value of the coefficient toward 0. This happens because (1) the predictor is not included in some models and, (2) because the weights are less than 1. This shrinkage is how model averaging accounts for model selection uncertainty.

If we carry out the model averaging for each of the coefficients, we find:

$$\begin{aligned}\hat{\beta}_0 &= 34.42(0.505) + 34.46(0.223) + 34.68(0.156) + 34.15(0.087) + 33.78(0.029) = 34.424 \\ \hat{\beta}_{\text{SES}} &= 3.241(0.505) + 3.237(0.223) + 3.223(0.156) + 3.321(0.087) + 3.393(0.029) = 3.248 \\ \hat{\beta}_{\text{GPA}} &= 5.064(0.505) + 5.050(0.223) + 5.028(0.156) + 5.169(0.087) + 5.304(0.029) = 5.071 \\ \hat{\beta}_{\text{Parent Education}} &= 0.4158(0.505) + 0.4143(0.223) + 0.4171(0.156) + 0.4114(0.087) + 0.4153(0.029) = 0.415 \\ \hat{\beta}_{\text{Locus-of-Control}} &= 1.369(0.505) + 1.165(0.223) + 1.263(0.156) + 0(0.087) + 0(0.029) = 1.147 \\ \hat{\beta}_{\text{Self-Esteem}} &= 0(0.505) + 0.3487(0.223) + 0.3767(0.156) + 0.9476(0.087) + 0(0.029) = 0.219 \\ \hat{\beta}_{\text{Locus-of-Control} \times \text{Self-Esteem}} &= 0(0.505) + 0(0.223) - 0.7216(0.156) + 0(0.087) + 0(0.029) = -0.112\end{aligned}$$

We can also obtain these values from the `model.avg()` function from the **MuMIn** package. This outputs two sets of model averaged coefficients. The output in the full row, corresponds to the values we just computed. (The weighted average is computed based on the full set of candidate models.)

```
# Model averaged estimates
model.avg(list(lm.1, lm.2, lm.3, lm.4, lm.5))
```

Call:

```
model.avg(object = list(lm.1, lm.2, lm.3, lm.4, lm.5))
```

Component models:

```
'1235' '12345' '123456' '1345' '135'
```

Coefficients:

```
(Intercept)      ses      gpa parent_ed      locus self_esteem
full          34.42402 3.248684 5.071313 0.4152767 1.147430  0.2186549
subset        34.42402 3.248684 5.071313 0.4152767 1.298565  0.4699563
      locus:self_esteem
full          -0.1122200
subset        -0.7215835
```

An alternative method of model averaging appears in the subset row. This *conditional averaging* only averages over the models where the parameter appears; it computes a weighted average over a subset of the candidate models. For example, the self-esteem predictor only appears in Models 4, 5, and 2, so the conditional weighted average would be computed as

$$\hat{\beta}_{\text{Self-Esteem}} = \frac{0.3487(0.223) + 0.3767(0.156) + 0.9476(0.087)}{0.223 + 0.156 + 0.087} = 0.4699$$

Using the full set of candidate models, the model averaging produces the following fitted model:

$$\begin{aligned} \widehat{\text{Achievement}}_i = & 34.42 + 3.25(\text{SES}_i) + 5.07(\text{Prior GPA}_i) + 0.42(\text{Parent Education}_i) + \\ & 0.22(\text{Self-Esteem}_i) + 1.15(\text{Locus-of-Control}_i) - 0.11(\text{Self-Esteem}_i)(\text{Locus-of-Control}_i) \end{aligned}$$

Estimates of the Unconditional Variance

Ideally, we would also like to obtain variance estimates for each of our coefficients. These typically give us an indication of the amount of sampling uncertainty in the coefficient. The variance estimates (or standard errors) we get from statistical output (e.g., `tidy()`) are *conditional on both the sample size and the model selected*. In other words, the numerical values for these estimates are computed using ML or OLS after the model and sample size are specified.

In a perfect world, we would use one set of data for the model selection process, and then use a second set of data (of identical sample size) to then obtain the coefficient and variance estimates by fitting the “best” model we previously identified. In practice, we tend to perform both model selection and estimation on the same data set. This, means that our variance estimates (which only assume sampling uncertainty), are too small, as they should be augmented to include model selection uncertainty in addition to the sampling uncertainty. Conceptually,

$$\text{Var}(\hat{\beta}_k) = \text{Sampling variance given a model} + \text{Variance due to model selection uncertainty}$$

Specifically, we can compute this updated variance estimate using,

$$\text{Var}(\hat{\beta}_k) = \sum w_i \left(\text{Var}(\hat{\beta}_{k_i} | \text{Model } i) + \left[\hat{\beta}_{k_i} - \hat{\beta}_k \right]^2 \right)$$

where

- w_i is the model probability (AICc weight) for Model i ,
- $\text{Var}(\hat{\beta}_{k_i} | \text{Model } i)$ is the sampling variance for $\hat{\beta}_{k_i}$ in Model i ,
- $\hat{\beta}_{k_i}$ is the estimated coefficient in Model i , and
- $\hat{\beta}_k$ is the model averaged value for $\hat{\beta}_k$ across all the candidate models.

The variance estimate is a weighted sum of two terms. The first term captures the sampling uncertainty for a particular term ($\hat{\beta}_k$) given that it is in Model i . The second term captures the model selection uncertainty by measuring how much that coefficient changes from one model to another. Note that when the coefficient doesn't change much from one model to another, this second term is small and the overall variance for $\hat{\beta}_k$ is essentially the conditional sampling variance.

If, however, the coefficient changes a great deal from model-to-model, omitting this last term (as we usually do when we only report standard errors from `tidy()`), underestimates the amount of uncertainty for the coefficient. This error propagates through the inference by underestimating the size of the p -values and confidence intervals (both should be larger).

Computing the Unconditional Variance for the Self-Esteem Coefficient

We can obtain the estimated coefficients and the sampling variances from the `tidy()` output for each model. Note that this output gives the conditional standard errors for the coefficients which can be squared to obtain the sampling variances.

Model	w_i	beta	var
1	0.029	0.0000	0.0000000
2	0.087	0.9476	0.2147396
3	0.505	0.0000	0.0000000
4	0.223	0.3487	0.3059196
5	0.156	0.3767	0.3063622

We can get the AICc weights from the

```
se
1 0.4627328
```

```
# Coefficient table using full set of models
coefTable(
  model.avg(list(lm.1, lm.2, lm.3, lm.4, lm.5)),
  full = TRUE
)
```

	Estimate	Std. Error
(Intercept)	34.42402	1.5719
ses	3.24868	0.5872
gpa	5.07131	0.4477

parent_ed	0.41528	0.2416
locus	1.14743	0.6607
self_esteem	0.21865	0.4630
locus:self_esteem	-0.11222	0.3609

References