Fitting and Interpreting a Proportional Odds Model

Take a look at the following table. It is a cross tabulation of data taken from the 1991 General Social Survey that relates political party affiliation to political ideology. (Agresti, *An Introduction to Categorical Data Analysis*, 1996)

Political Ideology by Party Affiliation, from the 1991 General Social Survey

	Very Liberal	**Slightly Liberal**	**Moderate**	**Slightly Conservative**	**Very Conservative**
Republican	30	46	148	84	99
Democratic	80	81	171	41	55

Not surprisingly, as we move across the table from left to right, response numbers for Democrats go down while those for Republicans go up. What if we wanted to model the probability of answering a particular political ideology given party affiliation? Since the political ideology categories have an ordering, we would want to use ordinal logistic regression.

There are several types of ordinal logistic regression models. Probably the most frequently used in practice is the proportional odds model. (Hosmer and Lemeshow, *Applied Logistic Regression* (2nd ed), p. 297)

Before we explain a “proportional odds model”, let’s just jump ahead and do it. Below we enter the data (since we don’t have the electronic source) and fit a proportional odds model using R:

```
# Table 8.6, Agresti 1996
party <- factor(rep(c("Rep","Dem"), c(407, 428)),
                levels=c("Rep","Dem"))
rpi <- c(30, 46, 148, 84, 99) # cell counts
dpi <- c(80, 81, 171, 41, 55) # cell counts
ideology <- c("Very Liberal","Slightly Liberal","Moderate","Slightly
Conservative","Very Conservative")
pol.ideology <- factor(c(rep(ideology, rpi),
                         rep(ideology, dpi)), levels = ideology)
dat <- data.frame(party,pol.ideology)

# fit proportional odds model
library(MASS)
pom <- polr(pol.ideology ~ party, data=dat)
```

Above we create a data frame with one row for each respondent. The first column we create is party, with 407 entries for Republican and 428 for Democratic. The next column we create is pol.ideology. We use the cell counts (stored as rpi and dpi, respectively) with the rep function to repeat each ideology a given number of times. For example, rep(ideology, rpi) repeats "Very Liberal" 30 times, "Slightly Liberal" 46 times, and so on. Finally we create a data frame called dat.

We fit the model using the `polr` function from the MASS package. "polr" stands for Proportional Odds Linear Regression. The MASS package comes with R. (Incidentally, MASS stands for *Modern Applied Statistics with S*, a book by W.N Venables and B.D. Ripley. R is an open-source implementation of S.)

Let's take a look at the model summary:

```
> summary(pom)

Call:
polr(formula = pol.ideology ~ party, data = dat)

Coefficients:
           Value Std. Error t value
partyDem -0.9745     0.1292  -7.545

Intercepts:
                                        Value    Std. Error t value
Very Liberal|Slightly Liberal            -2.4690   0.1318   -18.7363
Slightly Liberal|Moderate                -1.4745   0.1090   -13.5314
Moderate|Slightly Conservative            0.2371   0.0942     2.5165
Slightly Conservative|Very Conservative   1.0695   0.1039    10.2923

Residual Deviance: 2474.985
AIC: 2484.985
```

We have one coefficient and four intercepts. If you've never done ordinal logistic regression before, this may seem baffling and backwards. We'll explain in a moment. Let's move ahead with using our model to make predictions. Below we use our model to generate probabilities for answering a particular ideology given party affiliation:

```
> predict(pom,newdata = data.frame(party="Dem"),type="p")
         Very Liberal   Slightly Liberal       Moderate  Slightly
Conservative     Very Conservative
            0.1832505          0.1942837      0.3930552
0.1147559            0.1146547
> predict(pom,newdata = data.frame(party="Rep"),type="p")
         Very Liberal   Slightly Liberal       Moderate  Slightly
Conservative     Very Conservative
           0.07806044         0.10819225     0.37275214
0.18550357           0.25549160
```

The `newdata` argument requires data be in a data frame, hence the `data.frame` function. The `type="p"` argument says we want probabilities. The default is to return predicted class membership, which in this case would be "Moderate" since that's the highest estimated probability for both parties.

As far as R code goes, this is pretty simple. We fit a proportional odds model and got our estimated probabilities. But why four intercepts? What does the partyDem coefficient mean? Is this even an appropriate model? And why is this called "proportional odds"?

To answer these questions we need to state the proportional odds model:

$$logit[P(Y \leq j)] = \alpha_j - \beta x, j = 1, \ldots, J - 1$$

On the right side of the equal sign we see a simple linear model with one slope, β, and an intercept that changes depending on j, α_j. Here the j is the level of an ordered category with J levels. In our case, j = 1 would be "Very Liberal". So we see we have a different intercept depending on the level of interest. Why does j only extend to J – 1? Because in this model we're modeling the probability of being in one category (or lower) versus being in categories above it. In our example, $P(Y \leq 2)$ means the probability of being "Very Liberal" or "Slightly Liberal" versus being "Moderate" or above. Thus we're using the levels as boundaries. In this model the highest level returns a probability of 1 (i.e., $P(Y \leq J) = 1$), so we don't model it.

Now what about the logit? That means log odds. It's not the probability we model with a simple linear model, but rather the log odds of the probability. If probability is 0.75, the odds of success is 0.75/0.25 = 3. The log of 3 is about 1.09. So the logit of 0.75 is about 1.09.

The summary output of our model is stated in terms of this model. Since we have 5 levels, we get 5 – 1 = 4 intercepts. We have one predictor, so we have one slope coefficient. Plugging in values returns estimated log odds. Let’s try this. What’s the log odds a Democrat identifies as “Slightly Liberal” or lower? Plug in the appropriate values from the model output given above:

$$logit[P(Y \leq 2)] = -1.4745 – -0.9745(1) = -0.5$$

This isn’t terribly descriptive. Let’s convert to probability. The means taking the inverse logit. The formula for this is

$$P(Y \leq j) = \frac{exp(\alpha_j – \beta x)}{1 + exp(\alpha_j – \beta x)}$$

Applying to -0.5 we get

$$P(Y \leq 2) = exp(-0.5)/(1 + exp(-0.5)) = 0.378$$

This is cumulative probability. The probability of identifying as “Very Liberal” or “Slightly Liberal” when you’re a Democrat is about 0.378. This is why the labels for the intercepts in the summary output have a bar “|” between the category labels: they identify the boundaries.

So how did R calculate the probabilities for being in a particular category? In other words, how do we calculate $P(Y = j)$? We do some subtraction:

$$P(Y = j) = P(Y \leq j) – P(Y \leq j – 1)$$

For example the probability of a Democrat identifying as “Slightly Liberal” is

$$P(Y = 2) = P(Y \leq 2) – P(Y \leq 1) = 0.378 – 0.183 = 0.195$$

The baseline level in this model is Republican, so we set x = 0 when doing their calculations. The probability of a Republican identifying as “Slightly Liberal” or lower is simply

$$logit[P(Y \leq 2)] = -1.4745 – -0.9745(0) = -1.4745$$

$$P(Y \leq 2) = exp(-1.4745)/(1 + exp(-1.4745)) = 0.186$$

We can speed up these calculations by using elements of the `pom` object. The slope coefficient is stored in `pom$coefficient` and the intercepts are stored in `pom$zeta`. (The developers of the `polr` function like using zeta to represent the intercepts instead of alpha.) Here’s how to quickly calculate the cumulative ideology probabilities for both Democrats and Republicans:

```
# Democrat cumulative probabilities
> exp(pom$zeta - pom$coefficients)/(1 + exp(pom$zeta - pom$coefficients))
          Very Liberal|Slightly Liberal               Slightly 
Liberal|Moderate 
                              0.1832505 
0.3775341 
         Moderate|Slightly Conservative Slightly Conservative|Very 
Conservative 
                              0.7705894 
0.8853453 
# Republican cumulative probabilities
> exp(pom$zeta)/(1 + exp(pom$zeta))
          Very Liberal|Slightly Liberal               Slightly 
Liberal|Moderate 
                             0.07806044 
0.18625269 
         Moderate|Slightly Conservative Slightly Conservative|Very 
Conservative 
                             0.55900483 
0.74450840
```

That hopefully explains the four intercepts and one slope coefficient. But why the name “proportional odds”? “Proportional” means that two ratios are equal. Recall that odds is the ratio of the probability of success to the probability of failure. In this case, “success” and “failure” correspond to $P(Y \leq j)$ and $P(Y > j)$, respectively. The ratio of those two probabilities gives us odds. We can quickly calculate the odds for all J-1 levels for both parties:

```
# Democrat cumulative probabilities
dcp <- exp(pom$zeta - pom$coefficients)/(1 + exp(pom$zeta -
pom$coefficients))
# Republican cumulative probabilities
rcp <- exp(pom$zeta)/(1 + exp(pom$zeta))

# Democrat odds
> (dcp/(1-dcp))
          Very Liberal|Slightly Liberal               Slightly
Liberal|Moderate
                              0.2243656
0.6065138
         Moderate|Slightly Conservative Slightly Conservative|Very
Conservative
                              3.3589956
7.7218378
# Republican odds
> (rcp/(1-rcp))
          Very Liberal|Slightly Liberal               Slightly
Liberal|Moderate
                              0.0846698
0.2288827
         Moderate|Slightly Conservative Slightly Conservative|Very
Conservative
                              1.2675986
2.9140230
```

Now let's take the ratio of the Democratic ideology odds to the Republican ideology odds:

```
> (dcp/(1-dcp))/(rcp/(1-rcp))
          Very Liberal|Slightly Liberal               Slightly
Liberal|Moderate
                               2.649889
2.649889
         Moderate|Slightly Conservative Slightly Conservative|Very
Conservative
                               2.649889
2.649889
```

Look, they're all the same. The odds ratios are equal, which means they're proportional. And now we see why we call this a proportional odds model. For any level of ideology, the estimated odds that a Democrat's response is in the liberal direction (to the left) rather than the conservative direction is about 2.6 times the odds for republicans. Or to put it more succinctly, Democrats have higher odds of being liberal. News flash! But seriously, that's how you interpret odds ratios. Less than 1 means lower odds. More than 1 means higher odds. Since the baseline level of party is Republican, the odds ratio here refers to Democratic.

Let's take the log of the odds ratios:

```
> log((dcp/(1-dcp))/(rcp/(1-rcp)))
          Very Liberal|Slightly Liberal               Slightly
Liberal|Moderate
                              0.9745178
0.9745178
         Moderate|Slightly Conservative Slightly Conservative|Very
Conservative
                              0.9745178
0.9745178
```

Does that number look familiar? It's the slope coefficient in the model summary, without the minus sign. Some statistical programs, like R, tack on a minus sign so higher levels of predictors correspond to the response falling in the higher end of the ordinal scale. If we exponentiate the slope coefficient as estimated by R, we get exp(-0.9745) = 0.38. This means the estimated odds that a Democrat's response in the conservative direction (to the right) is about 0.38 times the odds for Republicans. That is, they're less likely to have an ideology at the conservative end of the scale. This isn't exactly a ground-breaking political discovery, but we have somewhat quantified the relationship between political ideology and party affiliation (at least as it existed in 1991).

When fitting a proportional odds model, it's a good idea to check the assumption of proportional odds. One way to do this is by comparing the proportional odds model with a multinomial logit model, also called an unconstrained baseline logit model. The multinomial logit model is typically used to model unordered responses and fits a slope to each level of the J – 1 responses. So whereas our proportional odds model has one slope coefficient and four intercepts, the multinomial model would have four intercepts and four slope coefficients. This suggests the proportional odds model is nested in the multinomial model, and that we could perform a likelihood ratio test to see if the models are statistically different. Here's how we can do that in R:

```
library(nnet)
mlm <- multinom(pol.ideology ~ party, data=dat)

M1 <- logLik(pom)
M2 <- logLik(mlm)
(G <- -2*(M1[1] - M2[1]))
[1] 3.687678
pchisq(G,3,lower.tail = FALSE)
[1] 0.2972241
```

First we load the nnet package, which has the `multinom` function for fitting multinomial logistic models. (The nnet package comes with R.) Then we calculate -2 times the difference between log likelihoods to obtain a likelihood ratio test statistic and save as G. Finally we calculate a p-value using the `pchisq` function, which tells us the area under a chi-square distribution with 3 degrees of freedom beyond 3.68. The p-value is quite high which indicates the proportional odds model fits as well as the more complex multinomial logit model. Hosmer and Lemeshow tell us this test is not “completely statistically correct.” Nevertheless it does “provide some evidence of model adequacy.” (page 304)

For questions or clarifications regarding this article, contact the UVa Library StatLab: statlab@virginia.edu (mailto:statlab@virginia.edu)

Clay Ford
Statistical Research Consultant
University of Virginia Library
October 5, 2015