Ordinal Outcomes Regression

Paul E. Johnson <pauljohn@ku.edu>

December 6, 2011

1 Introduction

This is my best effort to succinctly explain the theory behind the ordinal logistic regression model (with apologies to the probit model).

The main takeaway point is supposed to be this:

The same data leads to different estimates from different programs. That happens because the ordinal model can be written down in several different ways. None of them are wrong, but they are different, and as a result the user must be cautious.

Estimates obtained from four different programs are offered in Tables 2 through 4. If we line these up side by side, we see that estimates from one of the routines for R matches Stata (after chopping off the small differences in the decimals), while SAS appears to provide the "wrong sign" for the first row and the second procedure for R seems to provide the "wrong signs" for the second and third rows.

	R: polr	R: lrm	SAS	Stata
$\hat{b}_1$	-0.28	-0.28	0.28	-0.28
$\hat{\zeta}_1$	-4.24	4.24	-4.24	-4.24
$\hat{\zeta}_2$	-2.32	2.32	-2.32	-2.32

None of these are actually wrong, they are all correct *given the model they specified.* This the point at which the student may be tempted to give up. Please don't. I've worked very hard to clear this up in the following sections.

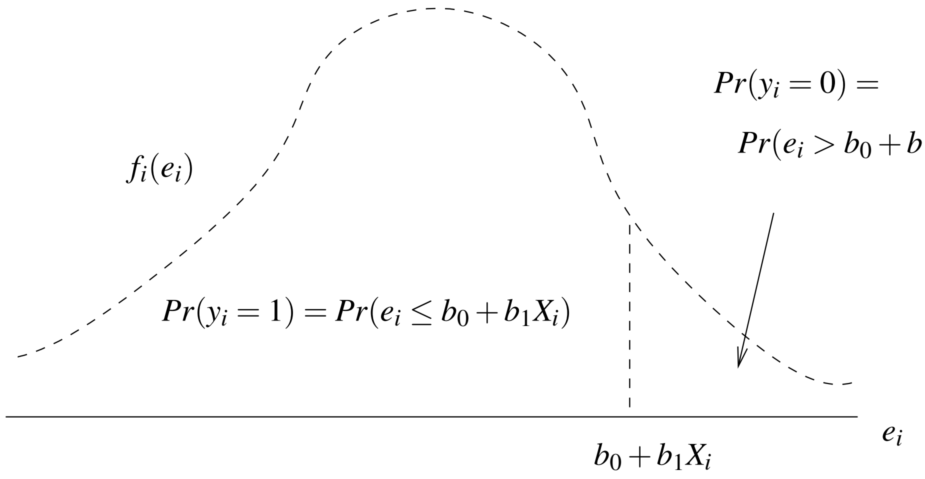
2 Extending the Logit Model to deal with Ordinal Dependent Variables

The easiest way to understand regression with ordinal dependent variables is to extend the "cumulative probability interpretation" of the two category model.

In the two category model, y_i is 1 with probability

$$F(b_0+b_1X_i)=\int_{-\infty}^{b_0+b_1X_i} f(e_i)de_i \tag{1}$$

Figure 1: Dichotomous Outcome Variable



And, of course, the probability that y_i is 0 will be $1 - F(b_0 + b_1X_i)$. The formula F is a "cumulative distribution function" (CDF), it represents the probability that a random variable e_i will be as small or smaller than $b_0 + b_1X_i$. The function f is a "probability density function" (PDF), which represents the probability that e_i is equal to some particular value. This is illustrated in Figure 1. The "probability density function" f is defined from left to right and the possible outcomes are divided into two sets by the line drawn at $e_i = b_0 + b_1X_i$. The area under the curve on the left side is the probability of getting a "yes" (or 1). The area on the right is the chance of a "no" (0).

Suppose y_i can have 3 values, 0, 1, and 2. (Keep in mind that this model can be written down in several ways. We tackle my favorite first, and then consider the others.) Leave the predictive part of the model $(b_0 + b_1X_i)$ the same, but we now introduce two new positive constants (Π_0 and Π_1) that divide the space. Considering Figure 2, it should be easy to see why some people call these new parameters "thresholds".

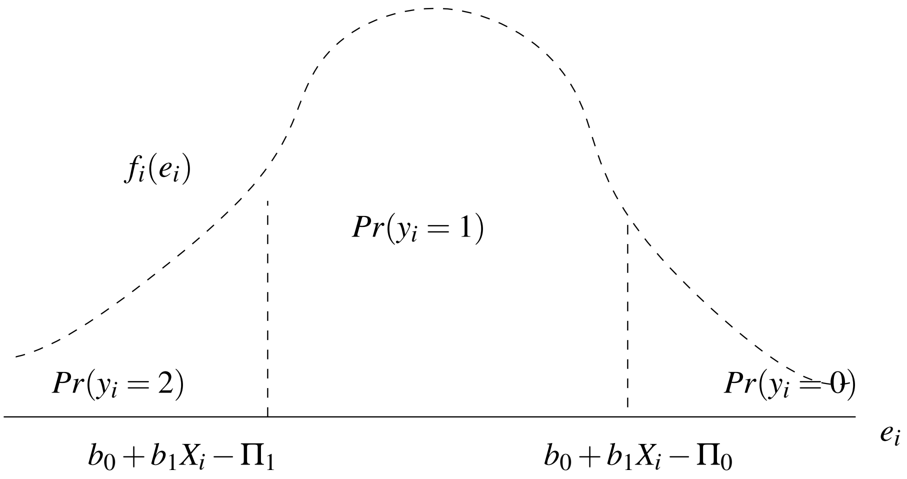
To summarize the effect of these new thresholds, we write down 1 equation for each possible outcome. My tendency is to write the thresholds as positive values like so:

$$y_i = \begin{cases} 2 & if\, b_0 + b_1X_i - e_i \geq \Pi_1 \\ 1 & if\, \Pi_0 \leq b_0 + b_1X_i - e_i < \Pi_1 \\ 0 & if\, b_0 + b_1X_i - e_i < \Pi_0 \end{cases} \tag{2}$$

Note we don't really need 3 equations. If we have two, say $Pr(y_i = 0)$ and $Pr(y_i = 1)$, then the chance of ending up in the other category is $1 - Pr(y_i = 0) - Pr(y_i = 1)$.

In order to translate this into a model involving the cumulative probability distribution, re-arrange so that the random variable e_i is by itself.

Figure 2: Ordinal Logit



$$y_i = \begin{cases} 2 & if\, e_i \le b_0 + b_1X_i - \Pi_1 \\ 1 & if\, b_0 + b_1X_i - \Pi_1 < e_i \le b_0 + b_1X_i - \Pi_0 \\ 0 & if\, b_0 + b_1X_i - \Pi_0 < e_i \end{cases} \tag{3}$$

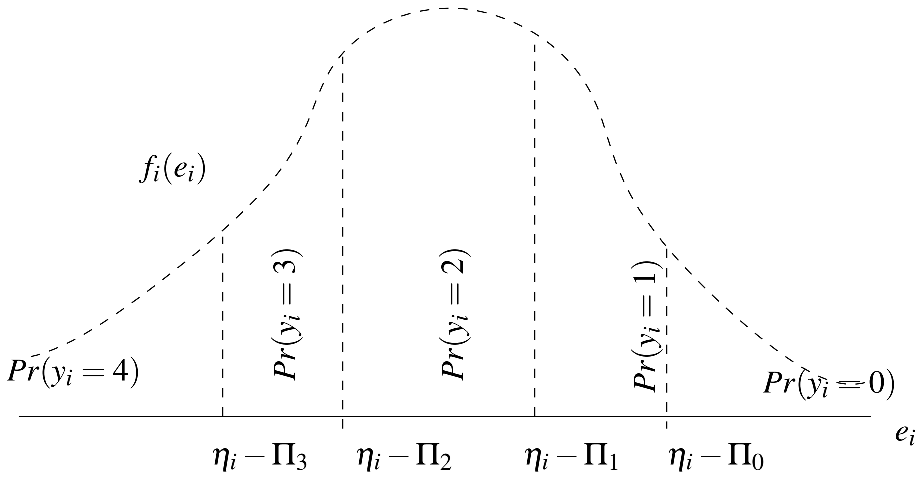
As in the dichotomous case, the probabilities of the various outcomes are calculated by use of cumulative probability. Rearrange 2 to convert these into probabilities of the individual outcomes.

$$\begin{aligned} Pr(y_i = 2) &= Pr(e_i \le b_0 + b_1X_i - \Pi_1) && = F(b_0 + b_1X_i - \Pi_1) \\ Pr(y_i = 1) &= Pr(b_0 + b_1X_i - \Pi_1 \le e_i < b_0 + b_1X_i - \Pi_0) && \\ &= 1 - F(b_0 + b_1X_i - \Pi_0) - F(b_0 + b_1X_i - \Pi_1) && \\ Pr(y_i = 0) &= Pr(b_0 + b_1X_i - \Pi_0 < e_i) && = 1 - F(b_0 + b_1X_i - \Pi_0) \end{aligned} \tag{4}$$

Note that any one category can be thought of as a “residual” category after the others have been assigned their shares. The middle category, $y_i = 1$, is left over if we “chop off” the outcomes on the left ($y_i = 2$) and the right ($y_i = 0$). We are left with the chance of ending up in the middle. In that sense, the probability of landing in the middle is equal to 1.0 minus the chance of a very small amount of random noise ($e_i \le b_0 + b_1X_i - \Pi_1$) and minus the chance of having a very large random noise ($b_0 + b_1X_i - \Pi_0 < e_i$). Similarly, the chances of being in the top category equal 1 minus the chance of ending up in the lower categories.

Any probability distribution can be used for the random error e_i, the two most common being Logistic and Normal. If the Normal is chosen, it is customary to call this a “probit” model and the symbol for the cumulative distribution is usually $\Phi()$.

Figure 3: Ordinal Model with More Categories



$$\eta_i = b_0 + b_1 X_i$$

What if your dependent variable have more categories? Add more thresholds! See the example in Figure 3.

3 Cumulative Probability Interpretation

3.1 The probability that y_i is greater than or equal to j.

In the previous section, I discussed the probability that $y_i = j$.

Another way to think of the same the model is to consider the probability that the observed outcome is in category j or in some "higher" category. That is, consider "cumulative probability," $Pr(y_i \geq j)$.

The cumulative model is used mainly because it effectively convers the multicategory outcome model into a sequence of 2 category comparisons. The probability model is somewhat easier to write down because we can use the familiar "logit" function.

Recall that, in the two category case, the logistic model is easily re-arranged so that the probability model

$$Pr(y_i = 1|X_i) = \frac{1}{1 + e^{-(b_0 + b_1 X_i)}} \tag{5}$$

becomes

$$ln\left[\frac{Pr(y_i = 1|X_i)}{1 - Pr(y_i = 1|X_i)}\right] = b_0 + b_1 X_i \tag{6}$$

That is to say, the “log of the odds ratio” is a linear function of the unknown coefficients and the observed variables. Because the right hand side “looks like an ordinary regression equation,” many people are more comfortable interpreting input variables as causing the change in the log of the odds ratio.

Suppose instead we re-write down the probability that y_i is greater than or equal to 1? (Pay no attention to the fact that y_i can only be 0 or 1 for a moment.) Then the dichotomous equation would be the same, except that we now use “greater than or equal to” rather than “equal to”:

$$ln\left[\frac{Pr(y_i \geq 1|X_i)}{1-Pr(y_i \geq 1|X_i)}\right] = b_0 + b_1 X_i. \tag{7}$$

For a three category model, it is obvious that the probability that y_i is 0 or greater is 1.0. The chance that y_i is greater than or equal to 1, or 2, is

$$\begin{array}{lll} Pr(y_i \geq 1) & = Pr(e_i \leq b_0 + b_1 X_i - \Pi_0) & = F(b_0 + b_1 X_i - \Pi_0) \\ Pr(y_i \geq 2) & = Pr(e_i \leq b_0 + b_1 X_i - \Pi_1) & = F(b_0 + b_1 X_i - \Pi_1) \end{array} \tag{8}$$

That should provide the intuition that we can add more categories and then use a sequence equations to represent a sequence of “steps,” or ordered comparisons. Is y_i greater than or equal to 1? Is y_i greater than or equal to 2?

If there are three outcome categories, we can write down two equations, one for the probability that y_i is greater than or equal to 1, and an second for the probability that it is greater than or equal to 2.

$$ln\left[\frac{Pr(y_i \geq 1)}{1-Pr(y_i \geq 1)}\right] = b_0 + b_1 X_i - \Pi_0 \tag{9}$$

$$ln\left[\frac{Pr(y_i \geq 2)}{1-Pr(y_i \geq 2)}\right] = b_0 + b_1 X_i - \Pi_1 \tag{10}$$

More succinctly, for k categories, we would write

$$for\, j \in \{1,\ldots,k\},\, ln\left[\frac{Pr(y_i \geq j)}{1-Pr(y_i \geq j)}\right] = b_0 + b_1 X_i - \Pi_{(j-1)} \tag{11}$$

By making the outcome of interest into a two category comparison–”bigger than j” or “not bigger than j”, there is a sense in which we have changed this into a series of two-category logistic regressions.

Of course, this is all consistent with the idea that the chances of observing an outcome of j or higher is

$$Pr(y_i \geq j) = \frac{1}{1+e^{-(b_0+b_1 X_i - \Pi_{(j-1)})}} \tag{12}$$

3.2 Cumulative Probability Implies Parallel Lines.

The basic ordinal logit model is often called the “parallel lines” model. Equations 9 and 10 are parallel lines–they have the same slope. Hence, a figure displaying lines for the log of the odds ratio would have parallel lines.

This makes the parallel lines assumption painfully obvious. The log odds that the outcome is in a particular category, or higher, depend on characteristics that are considered fixed (from the individual respondent's point of view) like b_0, b_1 and Π_j. To explain the differences among respondents, our attention should focus on X_i, which is proportional to the log odds ratio.

Another way to think of this is to compare the relative positions of the "dividing" lines in Figure 2. The position of the dividing lines between categories depends only on the differences in Π_j. The "linear predictor", $\eta_i = b_0 + b_1 X_i$, is the same in all dividers. The slope of b_1 summarizes the effect of X_i in the model. The equations that refer to the probability that a respondent will fall into a lower or higher category are parallel lines. A change in the predictor X_i has a proportional impact on η_i, which is then transformed into a statement about probability.

It is not actually necessary to assume that the slope b_1 is a constant when moving up the scale. There are implementations of the "non parallel lines" model. There will be $k - 1$ separate slope coefficients. The parameters proliferate as more predictors are added in the model. Estimating the extra parameters required by a non-parallel lines model makes this a messy numerical problem, one for which there will be no stable maximum likelihood estimates.

Most practitioners start with the parallel assumption, they may test it, and pursue the alternative only when they are persuaded it is truly necessary. If the lines are truly not parallel, then at some point one has to start to question the idea that the dependent variable is an ordinal indicator, because nonparallel lines do cross in a way that may change the ordering of the outcomes.

4 Sources of Confusion and Mis-Communication

There are many sources of possible confusion in this setup. I know of at least two dissertation chapters that came out completely wrong because of a mismatch between the software's design and the author's understanding.

Here are the most frequent sources of confusion.

1. In my Figure 2, the ordinal outcomes "step down" from left to right. Some models will code coefficients so that outcomes increase from left to right. That has the effect of reversing the sign of parameter estimates for b_1.

2. Some authors think of the combined effect of $b_0 + b_1 X_i \pm e_i$ as the random variable of interest, so the dividing points for that model will include just the Π_j and we think of the error adjusted linear predictor value moving among categories.

3. It is not possible to estimate Π_0, Π_1, and b_0 at the same time. One of those has to be "fixed" at 0 in order to allow the others to be estimated. It does not matter which one is fixed, the others will adjust. Some programs fix the separation threshold between the lowest value and the second lowest value at 0 and then provide estimates of b_0 and the other Π's. Some programs set the constant b_0 at 0 and then estimate all of the threshold parameters.

4. It may be that Π_0 or Π_1 are thought of as additional "constants" and their signs might be positive or negative. Where I have started with a positive threshold parameter Π_j like so:

$$b_0 + b_1 X_i + e_i \leq \Pi_j \tag{13}$$

some people prefer to think of Π_j as an adjustment in the intercept.

$$e_i \leq b_0 + b_1 X_i - \Pi_j = (b_0 - \Pi_j) + b_1 X_i, \tag{14}$$

one could as well reverse the scale of the thresholds

$$e_i \leq b_0 + b_1 X_i + \Pi_j \tag{15}$$

Recognizing the fact that we cannot estimate both b_0 and Π_0, some models will simply combine those two values from the beginning. That is, treat $\eta_1 = b_0 - \Pi_0$ as a combined "constant" ζ_1 which usually appears either as a constant in the predictive part of the model (and there is not separate "threshold" or separator estimated):

$$\zeta_1 + b_1 X_i \leq e_i \tag{16}$$

or as a new threshold in a model in which there is no "intercept" term in the predictive part.

$$b_1 X_i + e_i < \zeta_1 \tag{17}$$

5 RTFM: Read the Fine Manual

As I warned in a previous section, the ordinal logistic regression model can be written down in many ways. This is sometimes a blessing, sometimes a curse. It certainly forces us to think harder on the question of what we are estimating.

Over the years, I've had the same conversation with many different students. Here is a dramatic re-interpretation.

> Student: Here's an ordinal logit model output. What do these theta parameters mean?
>
> Professor: How should I know? Why did you choose to estimate them?
>
> Student: I didn't choose them. They just "came out."
>
> Professor: Go read the manual for that program you chose.
>
> Student: I didn't choose this program. I just Googled "ordinal logit" and some guy in Australia said I should do this.
>
> Professor: Better read the manual that goes along with that software.
>
> Student: But this other guy in Brazil says the thetas are probability numbers. What does that mean?
>
> Professor: Hm. Lets get the manual for this software you are using and find out what they are estimating. I'll read it to you.
>
> Student: Thanks, that's what I had been hoping for.

Table 1: Economic Expectations and Political Ideology: A Crosstabulation

```
> with(nes2002, prop.table(table(economy,conservatism), margin = 2))

        conservatism
economy           1          2          3          4          5          6
  better 0.00000000 0.03867403 0.05185185 0.01775148 0.05376344 0.08280255
  same   0.17391304 0.08839779 0.11851852 0.23076923 0.23655914 0.26114650
  worse  0.82608696 0.87292818 0.82962963 0.75147929 0.70967742 0.65605096
        conservatism
economy           7
  better 0.09375000
  same   0.28125000
  worse  0.62500000
```

Conservatism. We hear a lot of talk these days about liberals and conservativ

Economy. Now thinking about the economy in the country as a whole,
    would you say that over the past year the nation's economy has
    gotten BETTER, STAYED ABOUT THE SAME, or gotten WORSE?

In order for the reader to appreciate the fact that my drama is based on "true life" experience, I offer four sets of estimates from commonly used statistical procedures in Tables 2 (polr), 5 (lrm),3 (SAS), and 4 (Stata).

All of these estimates are based on the same data from the National Election Study of 2002. These models explore the effect of the respondent's political ideology on the respondent's opinion about the economy. I did not invest much thought in this choice of variables; there is no deep theory here. The model asserts that people who are conservative may have a different economic evaluations than people who are liberal. Since the president was a Republican at the time, my guess was that people who liked the conservative president were more likely to say the economy has gotten better. Political ideology is measured by the usual self-placement along a scale from "extremely liberal" to "extremely conservative", while the outcome variable as the respondents declare whether they think that the nation's economy has been "better", the same, or "worse" during the last year. A crosstabulation of the respondents is presented in "column proportion form" in Table 1.

6 polr

The R package that is distributed in support of Venables and Ripley's *Modern Applied Statistics* is called MASS. In that package, there is a routine called "polr" which stands for "proportional odds logistic regression." It can be used to fit ordinal logistic regression models. The threshold parameter $\mathbf{\Pi}_j$ in my treatment is called zeta, ζ_j, in their manual.

The polr documentation says they are thinking of the range of observed outcomes as a

Table 2: Output from MASS polr

```
> library (MASS)
> polr1 <- polr(economy~conservatism, data=nes2002, Hess=T)
> summary(polr1)

Call:
polr(formula = economy ~ conservatism, data = nes2002, Hess = T)

Coefficients:
               Value Std. Error t value
conservatism -0.2803    0.04502  -6.227

Intercepts:
            Value    Std. Error t value
better|same  -4.2441   0.2531   -16.7656
same|worse   -2.3215   0.2203   -10.5359

Residual Deviance: 1690.266
AIC: 1696.266
(270 observations deleted due to missingness)
```

result of dividing the real numbers like so: $(-\infty, \zeta_1, \zeta_2, ..., \zeta_K, +\infty)$. So the lowest categorical outcome is observed when the latent variable lies between $-\infty$ and ζ_1. The next value is observed when it is between ζ_1 and ζ_2, and so forth. What is the latent variable? The polr help page also provides the formula:

$$logit\, P(y_i \leq k|X_i) = \zeta_k - X_i b \tag{18}$$

In this context, $X_i b$ is the linear predictor, which is often called η_i. For example, $\eta_i = b_1 X1_i + b_2 X2_i$ (if there are 2 input variables). Note that there is no intercept coefficient b_0 in this formulation, it has been fixed at 0. That means we can estimate $k-1$ threshold-separating-parameters for a k category model.

A drawing of that particular parameterization is presented in Figure 4. The observed outcome categories are numbered $1, 2, \ldots, K$, and the k'th threshold separates the chance that $y_i \leq k$ from the chance that the outcome is greater than k. The model is designed so that the threshold separators ζ_k grow larger as k increases, so they have to be placed from left to right in the figure. Compared to my notation, this one has threshold parameters and with reversed signs.

The authors want to investigate the chances that an outcome is in a designated category k *or lower.* That means the probability model must be

$$Pr(y_i \leq k|X_i) = Pr(e_i < \zeta_k - X_i b) \tag{19}$$

The chance that the outcome will be k or smaller is equal to the chance that a random noise e_i is smaller than $\zeta_k - X_i b$. That, of course, is the same as the chance that the linear predictor $X_i b$ plus the random noise e_i remains below a threshold:

$$Prob(y_i \leq k|X_i) = Prob(X_i b + e_i < \zeta_k) \tag{20}$$

Figure 4: A polr Polychotomy

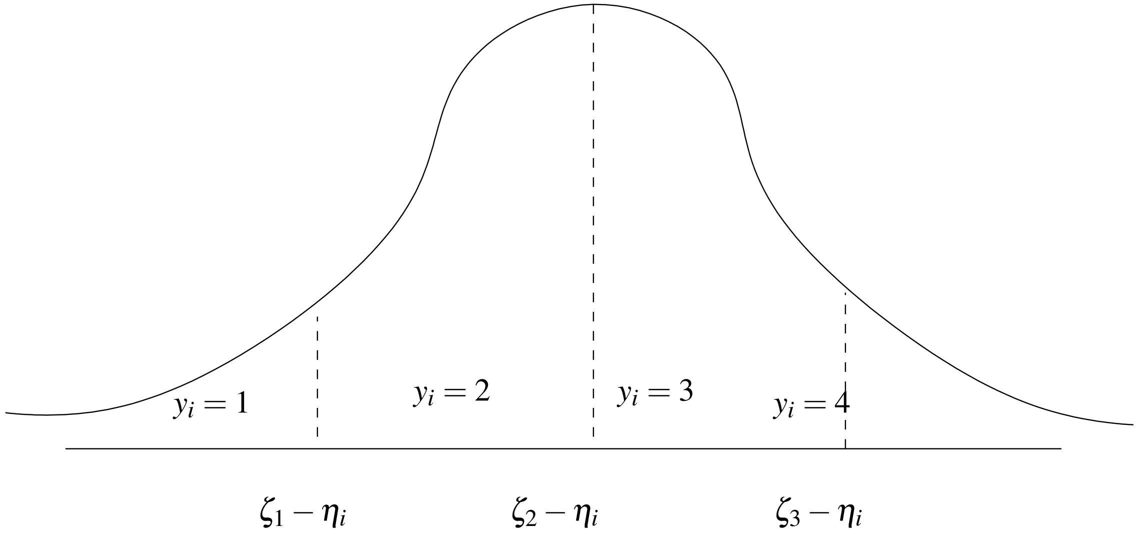
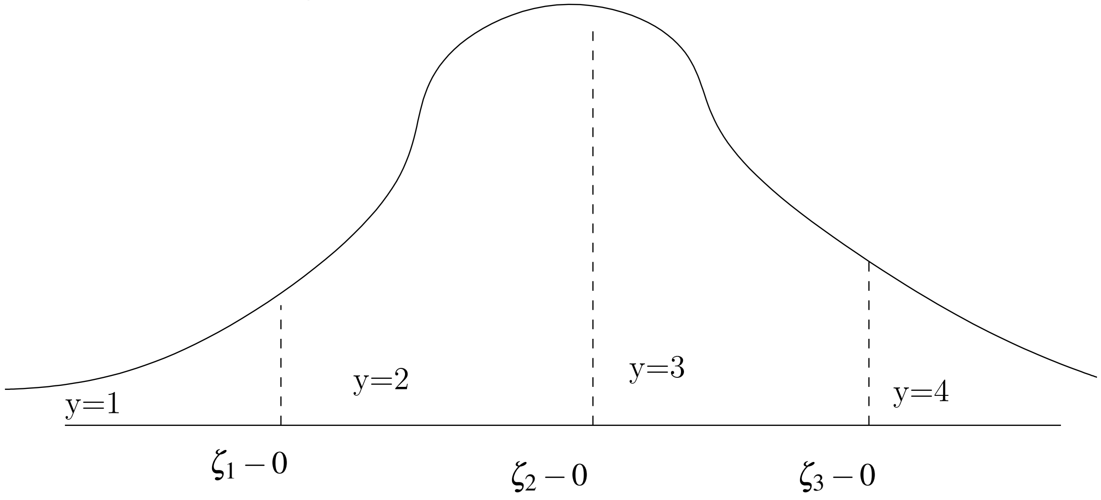


Figure 5: polr probabilities when $X_i b = 0$



The polr model, Venables and Ripley use a minus sign in equation 18. By writing 18 with a negative sign, it allows for the re-arranged model to have an intuitive interpretation. Simply put, if the linear predictor "gets bigger," then chance of a higher outcome is increased.

To illustrate that change, let's compare two cases.

1. Imagine $\eta_i = X_i b = 0$

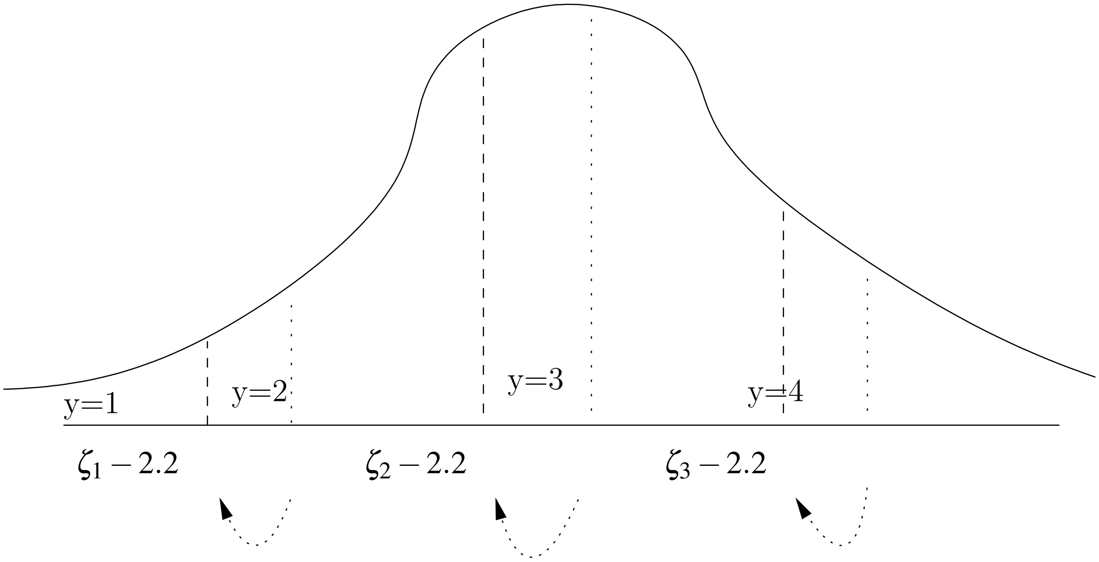
   If Xb were 0, what will be the probability distribution of outcomes? Then the probabilities would be represented in Figure 5

2. Imaine $\eta_i = X_i b = 2.2$

   What if Xb is a positive number, say 2.2. Then the dividers in the distribution change in the way shown in Figure 6

   The probability of a low outcome is reduced, while the probability of the biggest outcome is increased.

Figure 6: Shift in Probabilities



SAS's Proc Logistic went in the opposite direction. The default coding went in the other direction, so that when there was a large, statistically significant coefficient, it indicated that the chance of a higher outcome was growing smaller.

7 SAS Backwards Logistic

In SAS PROC LOGISTIC, the parameter estimate for the slopes of input variables are of the opposite sign. Observe in Table 3, the parameter estimate is 0.2805.

Why does the SAS estimate come out with the opposite value? It is as simple as this. Where the polr model was estimating this expression,

$$logit\, P(y_i \leq k|X_i) = \zeta_k - X_i b \tag{21}$$

SAS is estimating this

$$logit\, Pr(y_i \leq k|X_i) = \zeta_k + X_i b \tag{22}$$

The sign on the coefficient b is reversed, so the estimates are reversed. The sign on the threshold/intercept estimates is the same.

SAS is the only program that I've tested in which the slope estimates are reversed. The other programs try to help users avoid the mistake that so many SAS users have made, thinking that the effects in their data are all opposite of their expectations. Most users like to think of the input variables as predictors that increase the chances of more highly ranked outcomes. In a political participation study, we might say "high" represents "did vote" while "low" represents "did not." A positive slope coefficient should indicate that the chance of voting is increased. If we are modeling a person's vote for a Republican candidate, we usually think of "high" meaning "yes," the respondent voted for a Republican.

This is only a matter of interpretation, however. The original SAS model was designed to think of the low outcome as the one to be preferred, as in "lacking tumors" or "no heart disease."

Table 3: SAS Logistic is Still Backwards

```
proc logistic data=nes2002;

   model economy = conservatism;
run;

            Testing Global Null Hypothesis: BETA=0

    Test                     Chi−Square       DF      Pr > ChiSq

    Likelihood Ratio            41.1011        1          <.0001
    Score                       39.4521        1          <.0001
    Wald                        39.2135        1          <.0001


            Analysis of Maximum Likelihood Estimates

                                         Standard        Wald
 Parameter               DF   Estimate      Error  Chi−Square  Pr > ChiSq

 Intercept      better    1    −4.2458     0.2526    282.5931      <.0001
 Intercept      same      1    −2.3226     0.2196    111.9058      <.0001
 conservatism             1     0.2805     0.0448     39.2135      <.0001
```

In more recent versions of the SAS program, PROC LOGISTIC has been generalized to allow users to reverse the signs by the insertion of an option called DESCENDING.

8 Stata has the Slope We Expect, but Reversed Thresholds

In polr and PROC LOGISTIC, the "intercept" interpretation for the category separators was used. We think of the separation of the categorical outcomes as being driven by the replacement of b_0 with a sequence of separating constants.

In Stata, those separator constants are given the "threshold" interpretation, and thus their effects are reversed. See Table 4. The Stata manual explains their version of the model (I've reversed the roles of i and j to be consistent with the rest of this essay). "The probability of observing outcome j corresponds to the probability that the estimated linear function, plus random error, is within the range of the cutpoints estimated for the outcome:

$$Pr(outcome_i = j) = Pr(\kappa_{j-1} < \beta_1 x1_i + \beta_2 x2_i + \cdots + \beta_k xk_i + u_i \leq \kappa_j) \tag{23}$$

u_i is assumed to be logistically distributed in ordered logit" (STATABASE REFERENCE-MANUAL RELEASE 11, College Station, TX: Stata Press, 2009, p. 1268).

9 Goldilocks prefers lrm

I had not realized this until I was done with this exercise. The routine that provides parameter estimates of the divider coefficients and the slopes to match the theory I presented in Section 2 is Professor Frank Harrell's lrm estimator, which is part of a package that supports his fine book, *Regression Modeling Strategies*.

The output from lrm is presented in Table 5.

10 Calculating Predicted Probabilties

Recall that, while using the MASS package, the ordinal regression model is called a "proportional odds" model and the fitting function is called "polr". We previously used this fitting function:

```
polr1 <- polr(economy~conservatism, data=nes2002, Hess=T)
```

The function predict(polr1) returns a categorical output, the most likely outcome for each row of the data frame. This is a large data set, I won't print out all of the "Same" "Worse" or "Better" outcomes, but the tables don't hurt (much).

```
> newds <- polr1$model
> newds$polr1p1 <- predict(polr1)
> (t1 <- table(newds$polr1p1))
```

Table 4: Stata Ordinal Logistic Regression

```
. ologit economy conservatism

Iteration 0:   log likelihood = −865.68333
Iteration 1:   log likelihood = −845.38741
Iteration 2:   log likelihood = −845.13286
Iteration 3:   log likelihood = −845.13276
Iteration 4:   log likelihood = −845.13276

Ordered logistic regression                            Number of obs
=        1241
                                                       LR chi2(1)

=       41.10
                                                       Prob > chi2

=      0.0000
Log likelihood = −845.13276                            Pseudo R2
=      0.0237

------------------------------------------------------------------------------
     economy |    Coef. Std.Err.   z    P>|z|   [95% Conf. Interval]
-------------+----------------------------------------------------------------
conservatism | −.2805    .0450   −6.23  0.000   −.3687751     −.192286
-------------+----------------------------------------------------------------
       /cut1 | −4.2458   .2532                  −4.742078    −3.749548
       /cut2 | −2.3226   .2204                  −2.754558     −1.89068
------------------------------------------------------------------------------
```

Table 5: Output Harrell's lrm

```
> lrm1 <- lrm(economy~conservatism, data=nes2002)
> lrm1

Logistic Regression Model

lrm(formula = economy ~ conservatism, data = nes2002)

Frequencies of Missing Values Due to Each Variable
     economy conservatism
           9          266


                      Model Likelihood     Discrimination    Rank Discrim.
                         Ratio Test           Indexes           Indexes
Obs          1241    LR chi2      41.10    R2        0.043    C       0.611
 better        62    d.f.             1    g         0.484    Dxy     0.222
 same         258    Pr(> chi2) <0.0001    gr        1.623    gamma   0.277
 worse        921                          gp        0.022    tau-a   0.090
max |deriv| 4e-11                          Brier     0.047

             Coef    S.E.   Wald Z Pr(>|Z|)
y>=same       4.2458 0.2532 16.77  <0.0001
y>=worse      2.3226 0.2204 10.54  <0.0001
conservatism -0.2805 0.0450 -6.23  <0.0001
```

```
better   same  worse
     0      0   1241
```

```
> prop.table(table(newds$polr1p1, newds$conservatism), 2)

         1 2 3 4 5 6 7
  better 0 0 0 0 0 0 0
  same   0 0 0 0 0 0 0
  worse  1 1 1 1 1 1 1
```

As you can see, this model is a bit disappointing. This is not an uncommon problem. It predicts that every single respondent will say “worse”. In the data, the counts of “worse”,”same”, and “better” are 921, 258, and 62. The proportion of people predicted to say “worse” is 1.0 in every column.

The cross tabulation of the observed and predicted outcomes is often used to create a “box score” of the model’s predictions. This one predicts 921/1241 correctly, and 320/1240 incorrectly. The model predicts 74.2% correctly, but it is really quite a shallow victory. You could have predicted just as well without doing the regression at all. That’s why “percent correctly predicted” is not a powerful indicator of a model’s quality.

```
> table(newds$polr1p1, newds$economy)

         better same worse
  better      0    0     0
  same        0    0     0
  worse      62  258   921
```

Adding the option type=”p” indicates that we want the probabilities for the 3 categories to be returned in columns. Lets just inspect the first 5 rows of the new data set.

```
> newds$polr1p2 <- predict(polr1, type="p")
> colnames(newds)

[1] "economy"      "conservatism" "polr1p1"      "polr1p2"

> newds[1:5, ]

  economy conservatism polr1p1 polr1p2.better polr1p2.same polr1p2.worse
1   worse            3   worse     0.03219850   0.15314825    0.81465325
3   worse            4   worse     0.04217733   0.18926111    0.76856156
4  better            4   worse     0.04217733   0.18926111    0.76856156
5   worse            6   worse     0.07161613   0.27373170    0.65465216
6   worse            7   worse     0.09264175   0.31850556    0.58885269
```

I attempt to generate a bit of diversity in the predictions by throwing in more variables. Including additional variables, political party indicator variables(“repub” and “democ”), the “Bush Thermometer” scale and education (V023131) coded as a categorical predictor.

```
> library (MASS)
> polr2 <- polr(economy˜conservatism+repub+democ+V023010 + as.factor(V023131), data=ne
> summary(polr2)

Call:
polr(formula = economy ˜ conservatism + repub + democ + V023010 +
    as.factor(V023131), data = nes2002, Hess = T)

Coefficients:
                       Value Std. Error t value
conservatism        -0.04104   0.056812 -0.7224
repub               -0.17918   0.170964 -1.0481
democ                0.13717   0.194994  0.7035
V023010             -0.02636   0.003897 -6.7632
as.factor(V023131)2  0.89319   0.611484  1.4607
as.factor(V023131)3  1.17844   0.512134  2.3010
as.factor(V023131)4  1.48905   0.517943  2.8749
as.factor(V023131)5  1.32987   0.535315  2.4843
as.factor(V023131)6  1.54543   0.514715  3.0025
as.factor(V023131)7  1.40286   0.530757  2.6431

Intercepts:
            Value   Std. Error t value
better|same -3.7982  0.5999    -6.3310
same|worse  -1.7984  0.5906    -3.0448

Residual Deviance: 1576.397
AIC: 1600.397
(290 observations deleted due to missingness)
```

I have only meager success in diversifying my predictions.

```
> newds <- polr2$model
> newds$polr2p1 <- predict(polr2)
> (t2 <- table(newds$polr2p1))

better   same  worse
     0     10   1211

> table(newds$polr2p1, newds$economy)

         better same worse
  better      0    0     0
  same        4    4     2
  worse      57  249   905
```

```
> newds$polr2p2 <- predict(polr2, type="p")
> newds[1:5, ]

  economy conservatism repub democ V023010 as.factor(V023131) polr2p1
1   worse            3     0     1      50                  3   worse
3   worse            4     0     0      10                  4   worse
4  better            4     1     0      80                  6   worse
5   worse            6     1     0      80                  7   worse
6   worse            7     0     0      85                  4   worse
  polr2p2.better polr2p2.same polr2p2.worse
1    0.024774651  0.133234085   0.841991264
3    0.007694808  0.046484361   0.945820832
4    0.052561900  0.238126746   0.709311353
5    0.064942099  0.274140731   0.660917170
6    0.059543015  0.259116568   0.681340416

> library (MASS)
> nes2002$income <- factor(nes2002$V023149, levels=c("1","2","3","4","5","6","7"))
> nes2002$income <- as.numeric(levels(nes2002$income))[nes2002$income]
> nes2002$income[nes2002$income == 4] <- 3
> nes2002$income[nes2002$income == 5] <- 4
> nes2002$income[nes2002$income == 6] <- 5
> nes2002$income[nes2002$income == 7] <- 6
> table(nes2002$income)

  1   2   3   4   5   6
136 319 240 205 232 297

> race.black <- ifelse(nes2002$V023150 %in% c("1"),1,0)
> race.hispanic <- ifelse(nes2002$V023150 %in% c("4"),1,0)
> race.white <- ifelse(nes2002$V023150 %in% c("5"),1,0)
> table(race.black)

race.black
   0    1
1374  137

> table(race.white)

race.white
   0    1
 329 1182

> polr3 <- polr(economy~conservatism+repub+democ+V023010 + as.factor(V023131) + race.b
> summary(polr3)
```

```
Call:
polr(formula = economy ~ conservatism + repub + democ + V023010 +
    as.factor(V023131) + race.black + race.white + race.hispanic +
    income, data = nes2002, Hess = T)

Coefficients:
                       Value Std. Error  t value
conservatism        -0.03563   0.058472 -0.60931
repub               -0.24465   0.176288 -1.38778
democ                0.14637   0.202285  0.72360
V023010             -0.02621   0.004008 -6.53932
as.factor(V023131)2  0.73324   0.635919  1.15304
as.factor(V023131)3  0.89468   0.536265  1.66836
as.factor(V023131)4  1.18983   0.546028  2.17906
as.factor(V023131)5  1.08112   0.563720  1.91783
as.factor(V023131)6  1.16471   0.549783  2.11849
as.factor(V023131)7  1.03914   0.575012  1.80716
race.black           0.11498   0.387085  0.29704
race.white           0.37222   0.255449  1.45710
race.hispanic        0.03499   0.410892  0.08515
income               0.08941   0.048295  1.85129

Intercepts:
            Value   Std. Error t value
better|same -3.3941  0.6547    -5.1841
same|worse  -1.4246  0.6468    -2.2025

Residual Deviance: 1501.347
AIC: 1533.347
(354 observations deleted due to missingness)

>

> newds <- polr3$model
> newds$polr3p1 <- predict(polr3)
> (t3 <- table(newds$polr3p1))

better   same  worse
     0     10   1147

> table(newds$polr3p1, newds$economy)

         better same worse
  better      0    0     0
  same        4    4     2
  worse      56  235   856
```

```
> newds$polr3p2 <- predict(polr3, type="p")
> newds[1:5, ]

  economy conservatism repub democ V023010 as.factor(V023131) race.black
1   worse            3     0     1      50                  3          0
3   worse            4     0     0      10                  4          0
4  better            4     1     0      80                  6          1
5   worse            6     1     0      80                  7          0
6   worse            7     0     0      85                  4          0
  race.white race.hispanic income polr3p1 polr3p2.better polr3p2.same
1          0             0      5   worse     0.03033168   0.15279250
3          0             1      1   worse     0.01333862   0.07498924
4          0             0      5   worse     0.06680661   0.27227899
5          1             0      6   worse     0.05805119   0.24830991
6          1             0      4   worse     0.05538501   0.24048773
  polr3p2.worse
1    0.81687582
3    0.91167214
4    0.66091441
5    0.69363890
6    0.70412726
```

Well, supposing I did have to make something of this particular model, what is the next step? One option is to calculate the probability of a particular outcome for various combinations of the input variables. This can lead to a nice table. Lets look at the crosstabulation of income (on the 7 point scale) and conservatism (also on a 7 point scale). The distribution of respondents is:

```
> (t6 <- with(nes2002, table(conservatism, income )))

            income
conservatism  1  2  3  4  5  6
           1  2  6  4  4  3  2
           2 19 39 27 26 30 34
           3  7 23 19 21 30 30
           4 26 83 60 40 50 68
           5  3 21 31 30 31 63
           6 23 62 50 40 50 64
           7  5 10  9 13 11 12

> prop.table(t6, 2)

            income
conservatism           1           2           3           4           5
           1 0.023529412 0.024590164 0.020000000 0.022988506 0.014634146
           2 0.223529412 0.159836066 0.135000000 0.149425287 0.146341463
           3 0.082352941 0.094262295 0.095000000 0.120689655 0.146341463
           4 0.305882353 0.340163934 0.300000000 0.229885057 0.243902439
           5 0.035294118 0.086065574 0.155000000 0.172413793 0.151219512
           6 0.270588235 0.254098361 0.250000000 0.229885057 0.243902439
           7 0.058823529 0.040983607 0.045000000 0.074712644 0.053658537
            income
conservatism           6
           1 0.007326007
           2 0.124542125
           3 0.109890110
           4 0.249084249
           5 0.230769231
           6 0.234432234
           7 0.043956044
```