**Section 3.4 – Ordinal response models**

Suppose the response categories are ordered in the following way:

category 1 < category 2 < < category J

For example, a response variable may be measured using a Likert scale with categories strongly disagree, disagree, neutral, agree, or strongly agree. Logit transformations of the probabilities can incorporate these orderings in a variety of ways. In this section, we focus on one way where probabilities are cumulated based on these orderings.

The cumulative probability for Y is

P(Y j) = π1 + … + πj

for j = 1, …, J. Note that P(Y J) = 1. The logit of the cumulative probabilities can be written as



for j = 1, …, J – 1. For each j, we are computing the log odds of being in categories 1 through j vs. categories j + 1 through J.

When there is only one explanatory variable x, we can allow the log odds to vary by using a proportional odds regression model:



for j = 1, …, J – 1. Equivalently, the model is written as



The proportional odds name comes from there being no j subscripts on the β1 parameter, which means these regression parameters are the same for each possible log-odds that can be formed. This leads to each odds being a multiple of exp(βj0).

Notes:

* β10 < < βJ0 due to the cumulative probabilities. Thus, the odds increasingly become larger for j = 1, …, J – 1.
* It is o.k. to just refer to the model as a “proportional odds model” and drop the “regression” part.
* A proportional odds regression model actually is a special case of a cumulative probability regression model, which allows the regression parameter coefficient on each explanatory variable to vary as a function of j. We will examine this more general model later in this section.

For more than one explanatory variable, the model becomes:



for j = 1, …, J – 1.

What is πj only? Consider the case of one explanatory variable x again:



for j = 2, …, J – 1.

For j = 1,

π1 = P(Y = 1)

= P(Y 1) – P(Y 0)

= P(Y 1)

= .

For j = J,

πJ = P(Y = J)

= P(Y J) – P(Y J – 1)

= 1 – P(Y J – 1)

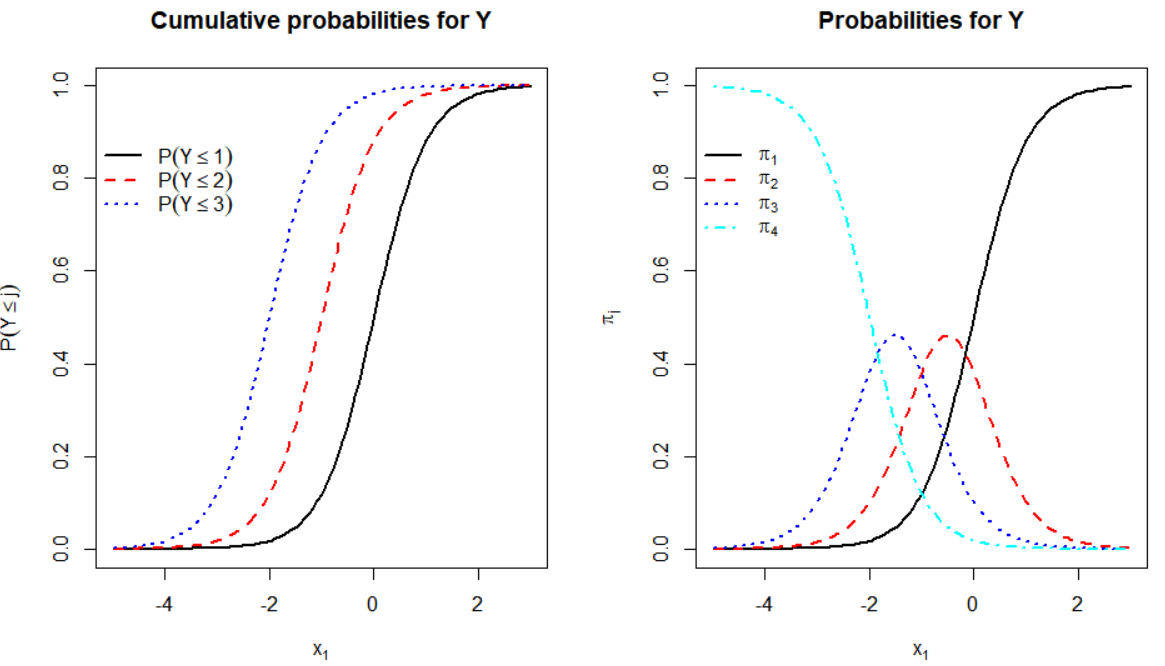
= .

Example: Proportional odds model plots (CumulativeLogitModelPlot.R)

The purpose of this example is to examine the shape of the proportional odds model. Consider the model



where β10 = 0, β20 = 2, β30 = 4, and J = 4. Through using the curve() function, below are plots of the model:



The plot on the left gives the cumulative probabilities:



The cumulative probability curves have the same shape, which is a result of β1 being shared for each response category j. The horizontal shift is due to the different values for βj0.

The plot on the right gives πj. We can see that particular response categories have larger probabilities than all of the other categories at specific values of x1.

Estimation and inference

Parameters are estimated using maximum likelihood estimation. For a sample of size m, the likelihood function is simply the product of m multinomial distributions with probability parameters πj. Iterative numerical procedures are used then to find the parameter estimates.

The polr() function from the MASS package (within the default installation of R) and the vglm() function from the VGAM package (need to download to install) are the two main ways to find the MLEs. Below are a few notes about these functions:

* polr()
  + This function is widely used because it is in the default installation of R.
  + The model form is slightly different than presented here. The upcoming example shows how a simple adjustment can be made to regression parameter estimates.
  + Profile LR inference procedures are available through the package’s method function for confint() when needed for simple regression parameters. The mcprofile package cannot be used.
  + Wald inference procedures are available through the emmeans package. This can be used for more complicated situations.
* vglm() – These are the same notes as given for multinomial regression.
  + This function is in a user-contributed package. While the package has been around for some time, it is much less used for these models.
  + The package's help warns “This package is undergoing continual development and improvement, therefore users should treat everything as subject to change.” This warning should not be taken lightly, because I have needed to change my own code when using this package due to its updates.
  + Profile LR intervals can be calculated for simple regression parameters using the package’s method function for confint(). However, this function cannot perform calculations for more complicated situations, like a linear combination of regression parameters. Also, the mcprofile package does not have the capability to find these intervals.
  + The emmeans package cannot be used with it.

Notes:

* For both functions, it is especially important to have the levels of the categorical response ordered in the desired way; otherwise, the ordering of the levels for Y will not be correctly taken into account.
* The covariance matrix for the parameter estimates follows from using standard likelihood procedures as outlined in Appendix B.
* Neither function and corresponding package is as good as I would like. In my notes, I will focus on polr() and provide additional content with vglm() as needed.

Hypotheses

It is important to make sure that one knows what the hypotheses represent in the context of the model. Suppose there is one explanatory variable (p = 1), and the hypotheses of interest are

H0: β1 = 0

Ha: β1 ≠ 0

If the null hypothesis is true, this says that the log-odds comparing P(Y ≤ j) to P(Y > j) do not depend on the explanatory variable. In the case of one categorical explanatory variable X, this is equivalent to independence between X and Y.

If the alternative hypothesis is true, the ordering of the log-odds comparing P(Y ≤ j) to P(Y > j) holds; i.e., the log-odds progressively grow larger or smaller depending on the sign of β1. Thus, you see the ordering of the πj values as shown in the proportional odds model plots.

Contrast the alternative hypothesis meaning to the corresponding test for the multinomial regression model. Suppose there is one explanatory variable (p = 1), and the hypotheses of interest are

H0: β21 = = βJ1 = 0

Ha: At least one βj1 ≠ 0

The alternative hypothesis does not say what type of trend (if any) exists among the log-odds ratios.

The fewer constraints for the multinomial regression model alternative hypothesis may seem good, but it leads to a model that is less efficient when the proportional odds assumptions are applicable. The efficiency loss will be evident with inferences. For example, the hypothesis test involving multinomial regression models will result in less power than the hypothesis test involving the proportional odds model

Example: Wheat kernels (Wheat.R, Wheat.csv)

The proportional odds model may be useful with this example. Overall, we would like to have a “healthy” kernel so healthy is greater than both sprout and scab. I would think that a sprout kernel would be preferable to a scab kernel. Thus, scab (Y = 1) < sprout (Y = 2) < healthy (Y = 3).

Here is how I reorder a factor in R to recognize these orderings:

> levels(wheat$type)

[1] "Healthy" "Scab" "Sprout"

> wheat$type.order <- factor(x = wheat$type, levels = c("Scab", "Sprout", "Healthy"))

> # head(wheat) #excluded to save space

> levels(wheat$type.order)

[1] "Scab" "Sprout" "Healthy"

I would like to estimate the following model:

 for j = 1, 2

and x1, …, x6 corresponding to the ordering of the explanatory variables in the data frame.

The proportional odds model is estimated using polr():

> library(package = MASS)

> mod.fit.ord <- polr(formula = type.order ~ class + density + hardness + size + weight + moisture, data = wheat, method = "logistic")

> summary(mod.fit.ord)

Re-fitting to get Hessian

Call: polr(formula = type.order ~ class + density + hardness + size + weight + moisture, data = wheat, method = "logistic")

Coefficients:

Value Std. Error t value

classsrw 0.17370 0.391764 0.4434

density 13.50534 1.713009 7.8840

hardness 0.01039 0.005932 1.7522

size -0.29253 0.413095 -0.7081

weight 0.12721 0.029996 4.2411

moisture -0.03902 0.088396 -0.4414

Intercepts:

Value Std. Error t value

Scab|Sprout 17.5724 2.2460 7.8237

Sprout|Healthy 20.0444 2.3395 8.5677

Residual Deviance: 422.4178

AIC: 438.4178

The actual model estimated by polr() is



where -ηr is βr in our notation. Thus, we will always need to change the sign of these estimated regression parameters given by polr(). The estimated model is:



where  and .

The “t value” column in the coefficients table provides the Wald statistic for testing H0: βr = 0 vs. Ha: βr ≠ 0 for r = 1, …, 6, and the Anova() function provides the corresponding LRTs:

> library(package = car) # If not done already

> Anova(mod.fit.ord)

Analysis of Deviance Table (Type II tests)

Response: type.order

LR Chisq Df Pr(>Chisq)

class 0.197 1 0.65749

density 98.437 1 < 2.2e-16 \*\*\*

hardness 3.084 1 0.07908 .

size 0.499 1 0.47982

weight 18.965 1 1.332e-05 \*\*\*

moisture 0.195 1 0.65872

---

Signif. codes: 0 '\*\*\*' 0.001 '\*\*' 0.01 '\*' 0.05 '.' 0.1 ' ' 1

Because of the small p-values for density and weight, there is sufficient evidence that these are important explanatory variables. Also, there is marginal evidence that hardness is important too.

The predict() function estimates the probabilities for each response category:

> pi.hat.ord <- predict(object = mod.fit.ord, type = "probs")

> head(pi.hat.ord)

Scab Sprout Healthy

1 0.03661601 0.2738502 0.6895338

2 0.03351672 0.2576769 0.7088064

3 0.08379891 0.4362428 0.4799583

4 0.01694278 0.1526100 0.8304472

5 0.11408176 0.4899557 0.3959626

6 0.02874814 0.2308637 0.7403882

> head(predict(object = mod.fit.ord, type = "class"))

[1] Healthy Healthy Healthy Healthy Sprout Healthy

Levels: Scab Sprout Healthy

For example, the estimated probability of being healthy for the first observation is



With respect to confidence intervals for πj, the MASS package (and VGAM package) do not provide ways to calculate them for the same reasons as for multinomial regression models. Below is emmeans code to calculate these intervals for the first observation.

> library(package = emmeans)

> calc.prob <- emmeans(object = mod.fit.ord, specs = ~ type.order, at = wheat[1,], mode = "prob")

Re-fitting to get Hessian

> summary(object = calc.prob, level = 0.95)

type.order prob SE df asymp.LCL asymp.UCL

Scab 0.0366 0.0156 Inf 0.00612 0.0671

Sprout 0.2739 0.0693 Inf 0.13797 0.4097

Healthy 0.6895 0.0820 Inf 0.52882 0.8502

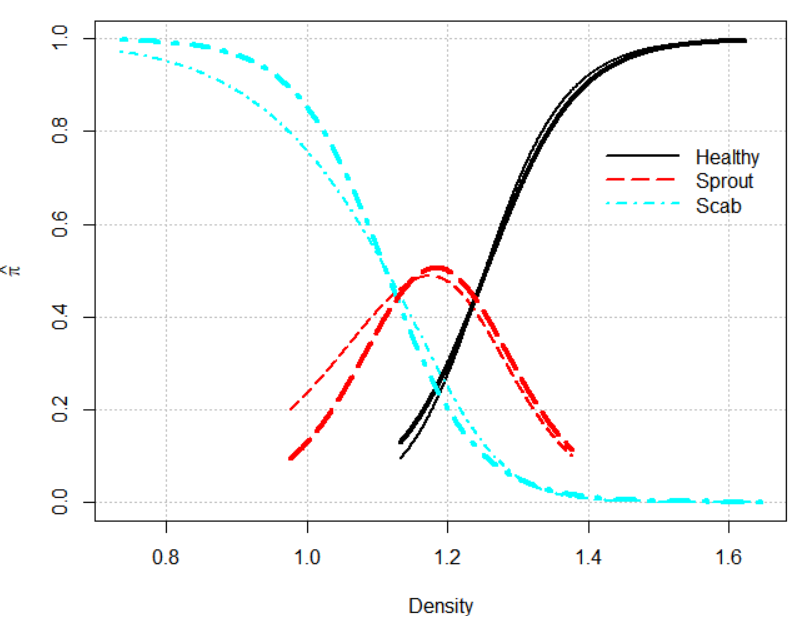
Confidence level used: 0.95

The argument values are similar to those used for the multinomial regression models.

When there is only one explanatory variable in the model, we can easily examine the estimated probabilities through a plot. The model using only density is



 and . Through using multiple calls to the curve() function (see program), I constructed the plot below, where the thicker lines are for the proportional odds model and the thinner lines are for the multinomial regression model.



Compare the results from this model to what we found earlier in the parallel coordinates plot.