**Chapter 3 – Analyzing a multicategory response**

The previous two chapters provided analysis methods for when there were binary responses. The purpose of this chapter is to generalize some of these previous methods to allow for more than two response categories. Examples include:

* Canadian political party affiliation – Conservative, New Democratic, Liberal, Bloc Quebecois, or Green
* Chemical compounds in drug discovery experiments – Positive, blocker, or neither
* Cereal shelf-placement in a grocery store – Bottom, middle, or top
* Beef grades – Prime, choice, select, standard, utility, and commercial
* Five-level Likert scale – Strongly disagree, disagree, neutral, agree, or strongly agree.

For these examples, some responses are ordinal (e.g., Likert scale) and some are not (e.g., chemical compounds). We will investigate both ordinal and nominal (unordered) multicategory responses within this chapter.

**Section 3.1 – Multinomial probability distribution**

The multinomial probability distribution is the extension of the binomial distribution to situations where there are more than two categories for a response.

Notation:

* Y denotes the response category with levels of j = 1, …, J
* Each category has a probability of πj = P(Y = j).
* n denotes the number of trials
* n1, …, nJ denote the response count for category j, where 

The probability mass function for observing particular values of n1, …, nJ is



The dmultinom() function can evaluate this function, and the rmultinom() function can simulate observations. When J = 2, the distribution simplifies to the binomial distribution.

For n trials, the likelihood function is simply the probability mass function. The maximum likelihood estimate of πj is .

Questions:

* What would the probability mass function look like if there was only one trial?
* What would the likelihood function be if 1 trial was observed, then another 1 trial was observed independently of the previous trial, … , so that there were m total one-trial sets observed in succession?
* What would the likelihood function be if n trials were observed, then another n trials were observed independently of the previous trial, … , so that there were m total n-trial sets observed in succession?
* Consider the same scenario as in the last question, but now with the possibility of different probabilities for each set. What would the likelihood function be?

Example: Multinomial simulated sample (Multinomial.R)

As a quick way to see what a sample looks like in a multinomial setting, consider the situation with n = 1,000 trials, π1 = 0.25, π2 = 0.35, π3 = 0.2, π4 = 0.1, and π5 = 0.1. Below is how we can simulate a sample:

> pi.j<-c(0.25, 0.35, 0.2, 0.1, 0.1)

> set.seed(2195) #Set a seed to be able to reproduce the

sample

> n.j<-rmultinom(n = 1, size = 1000, prob = pi.j)

> data.frame(n.j, pihat.j = n.j/1000, pi.j)

n.j pihat.j pi.j

1 242 0.242 0.25

2 333 0.333 0.35

3 188 0.188 0.20

4 122 0.122 0.10

5 115 0.115 0.10

Suppose there are m = 5 separate sets of n = 1000 trials.

> set.seed(9182)

> n.j<-rmultinom(n = 5, size = 1000, prob = pi.j)

> n.j

[,1] [,2] [,3] [,4] [,5]

[1,] 259 259 237 264 247

[2,] 341 346 374 339 341

[3,] 200 188 198 191 210

[4,] 92 106 89 108 107

[5,] 108 101 102 98 95

> n.j/1000

[,1] [,2] [,3] [,4] [,5]

[1,] 0.259 0.259 0.237 0.264 0.247

[2,] 0.341 0.346 0.374 0.339 0.341

[3,] 0.200 0.188 0.198 0.191 0.210

[4,] 0.092 0.106 0.089 0.108 0.107

[5,] 0.108 0.101 0.102 0.098 0.095

Notice the variability from one set to another.

**Section 3.2 – I×J contingency tables and inference procedures**

Section 1.2 introduced us to a 2×2 contingency table. Now, we look at how to extend this to an I×J contingency table. We will begin by focusing on two separate ways that one can think of how the counts arise in a contingency table structure through using a multinomial distribution. Chapter 4 will consider another way through using a Poisson distribution. Section 6.2 considers another way through using a hypergeometric distribution.

One multinomial distribution

Set-up:

* X denotes the row variable with levels i = 1, …, I
* Y denotes the column variable with levels j = 1, …, J
* P(X = i, Y = j) = πij
* 
* nij denotes the cell count for row i and column j
* 

Contingency tables summarizing this information are shown below:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Y | | | |  |
|  |  | 1 | 2 |  | J |  |
| X | 1 | 11 | 12 |  | 1J | 1+ |
| 2 | 21 | 22 |  | 2J | 2+ |
|  |  |  |  |  |  |
| I | I1 | I2 |  | IJ | I+ |
|  |  | +1 | +2 |  | +J | 1 |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Y | | | |  |
|  |  | 1 | 2 |  | J |  |
| X | 1 | n11 | n12 |  | n1J | n1+ |
| 2 | n21 | n22 |  | n2J | n2+ |
|  |  |  |  |  |  |
| I | nI1 | nI2 |  | nIJ | nI+ |
|  |  | n+1 | n+2 |  | n+J | n |

The set-up given for these contingency tables fits right into the multinomial setting of Section 3.1. We now just categorize the responses with respect to X and Y. The probability mass function for observing particular values of n11, …, nIJ is



The MLE of πij is the estimated proportion  = nij/n.

We can also discuss marginal distributions for X and for Y as well:

* X is multinomial with counts ni+ for i = 1, …, I and corresponding probabilities πi+. The maximum likelihood estimate of πi+ is  = ni+/n.
* Y is multinomial with counts n+j for j = 1, …, J and corresponding probabilities π+j. The MLE of π+j is  = n+j/n

Example: Multinomial simulated sample (Multinomial.R)

As a quick way to see what a sample looks like in a 2×3 contingency table setting, consider the situation with n = 1,000 observations, π11 = 0.2, π21 = 0.3, π12 = 0.2, π22 = 0.1, π13 = 0.1, and π23 = 0.1. Below is how we can simulate a sample:

> pi.ij<-c(0.2, 0.3, 0.2, 0.1, 0.1, 0.1)

> pi.table<-array(data = pi.ij, dim = c(2,3), dimnames =

list(X = 1:2, Y = 1:3))

> pi.table

Y

X 1 2 3

1 0.2 0.2 0.1

2 0.3 0.1 0.1

> set.seed(9812)

> save<-rmultinom(n = 1, size = 1000, prob = pi.ij)

> c.table1<-array(data = save, dim = c(2,3), dimnames =

list(X = 1:2, Y = 1:3))

> c.table1

Y

X 1 2 3

1 191 206 94

2 311 95 103

> c.table1/sum(c.table1)

Y

X 1 2 3

1 0.191 0.206 0.094

2 0.311 0.095 0.103

I multinomial distributions

Instead of using one multinomial distribution, one can think of the data arising through separate multinomial distributions for each row. Thus, there are I multinomial distributions. This can be thought of as a direct extension to what we had in Section 1.2 with two binomial distributions (one for each row).

Set-up:

* ni+ as fixed row counts
* P(Y = j | X = i) = πj|i represents the conditional probability of observing response category j given an item is in group i
* ni1, …, niJ are the counts with corresponding probabilities π1|i, …, πJ|i.
*  for i = 1, …, I

We can view the contingency table in terms of these conditional probabilities:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  |  | Y | | | |  |
|  |  | 1 | 2 |  | J |  |
| X | 1 | 1|1 | 2|1 |  | J|1 | 1 |
| 2 | 1|2 | 2|2 |  | J|2 | 1 |
|  |  |  |  |  |  |
| I | 1|I | 2|I |  | J|I | 1 |

The probability mass function is



The likelihood function is the same as this function, and the MLE of πj|i is . Notice how these estimates can be found from the previous MLEs in the one multinomial setting: .

Example: Multinomial simulated sample (Multinomial.R)

Consider again a 2×3 contingency table setting. Suppose π1|1 = 0.4, π2|1 = 0.4, π3|1 = 0.2, π1|2 = 0.6, π2|2 = 0.2, and π3|2 = 0.2. These conditional probabilities actually result from using πij in the previous example:

> pi.cond<-pi.table/rowSums(pi.table)

> pi.cond # pi\_j|i

Y

X 1 2 3

1 0.4 0.4 0.2

2 0.6 0.2 0.2

In the previous example, the row totals were random variables. Here, the row totals are fixed. Let n1+ = 400 and n2+ = 600. Below is how I simulate a sample:

> set.seed(8111)

> save1<-rmultinom(n = 1, size = 400, prob = pi.cond[1,])

> save2<-rmultinom(n = 1, size = 600, prob = pi.cond[2,])

> c.table2<-array(data = c(save1[1], save2[1], save1[2],

save2[2], save1[3], save2[3]), dim = c(2,3), dimnames =

list(X = 1:2, Y = 1:3))

> c.table2

Y

X 1 2 3

1 162 159 79

2 351 126 123

> rowSums(c.table2)

1 2

400 600

> c.table2/rowSums(c.table2)

Y

X 1 2 3

1 0.405 0.3975 0.1975

2 0.585 0.2100 0.2050

> round(c.table1/rowSums(c.table1),4)

Y

X 1 2 3

1 0.389 0.4196 0.1914

2 0.611 0.1866 0.2024

Independence

Consider the one multinomial distribution case again. Independence between X and Y when the occurrence of X = i does not have an effect of the occurrence of Y = j for each i = 1, …, I and j = 1, …, J. Symbolically, independence exists when

πij = πi+π+j

for i = 1, …, I and j = 1, …, J. In words, this means that the probability of an item being in cell (i,j) only involves knowing the separate marginal probabilities for X and for Y; thus, X and Y are independent of each other.

Why is independence important?

Independence helps to simplify the understanding of probabilities within the contingency table! There are only I – 1 + J – 1 = I + J – 2 unknown probability parameters. Note that the “-1” parts occur due to the  and .

Without independence, there are IJ – 1 unknown probability parameters, where the “-1” part occurs due to the . We will examine how to perform a hypothesis test for independence shortly.

Question: In the previous R one multinomial distribution example, how would you simulate a sample under independence?

Consider the I multinomial distributions case again. Similar to Section 1.2, it is often interest to know if these conditional probabilities are equal across the rows of the table. Thus, we want to know if πj|1 = = πj|I for j = 1, …, J. Note that this is mathematically equivalent to πij = πi+π+j for i = 1, …, I and j = 1, …, J!

From your first statistics course, you learned that . Thus,



under independence. Because each πj|i  is equal to π+j for each i, we have πj|1 = = πj|I.

Because of this equivalence, I will refer to πj|1 = = πj|I for j = 1, …, J as “independence” as well.

Question: In the previous R I multinomial distributions example, how would you simulate a sample under independence?

Test for independence

The hypotheses are:

H0: πij = πi+π+j for i = 1,…,I and j = 1,…,J

Ha: Not all equal

Remember that a Pearson chi-square test statistic calculates



for every cell of a contingency table and sums these quantities. The Pearson chi-square test for independence then uses the statistic



Notes:

* The estimated expected cell count is  = .
* X2 is equivalent to the corresponding statistic used in Section 1.2 for the Pearson chi-square test for a 2×2 contingency table.
* If the null hypothesis is true, X2 has a  distribution for a large sample.
* Reject the null hypothesis if .

The LRT statistic is formed the usual way with



The numerator of Λ uses  to estimate πij, and the denominator of Λ uses  to estimate πij. The transformed statistic simplifies to



where we use 0×log(0) = 0. The large sample distribution is the same as for X2.

Degrees of freedom

Where do the degrees of freedom come from for a test of independence?

A general way to find degrees of freedom for a hypothesis test is to calculate:

(Number of free parameters under Ha)

– (Number of free parameters under H0)

Under the alternative hypothesis for the test of independence, we have IJ πij parameters with the restriction that . When independence is true, we need the I different πi+ and the J different π+j parameters to find πij with the restriction that  and . Thus, the overall degrees of freedom is (IJ – 1) – (I + J – 2) = (I – 1)(J – 1).

“Large sample”

What is a large enough sample to have the  work well?

This is not an easy question to answer! A common recommendation is for ni+n+j/n > 1 or > 5 for all cells of the contingency table.

What if these recommendations are not satisfied?

A  distribution may not work!

How could this affect your hypothesis test decision?

When a distributional approximation is in doubt, there are a few things that can be done:

1. Use exact inference methods (Section 6.2)
2. Use Monte Carlo simulation (discussed soon!)

Example: Fiber enriched crackers (Fiber.R, Fiber.csv)

Fiber is often added to foods as a convenient way for people to consume it. The Data and Story Library (DASL) describes the results of a study where individuals are given a new type of fiber enriched cracker. The participants ate the crackers and then a meal. Shortly afterward, the participants were instructed to describe any bloating that they experienced. Below is the data:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  |  | Bloating severity | | | |
|  |  | None | Low | Medium | High |
| Fiber source | None | 6 | 4 | 2 | 0 |
| Bran | 7 | 4 | 1 | 0 |
| Gum | 2 | 2 | 3 | 5 |
| Both | 2 | 5 | 3 | 2 |

The purpose of this data is to determine if the fiber source has an effect on the bloating severity.

Before beginning the data analysis, below are a few notes:

* Notice the columns have ordinal levels. We will take this into account later in the chapter. It is instructive in a class setting to analyze the data first without taking the order into account, so that we can see the benefits of taking into account the order later.
* I would expect that each person fits in one and only one cell of the table. Why would this be important to know?
* Given the layout of the data, it is likely that the sample size for each row was fixed. Thus, this would correspond to the I multinomial distribution setting.
* Fiber source could actually be analyzed as two separate explanatory variables – bran (“yes” or “no”) and gum (“yes” or “no”). I am going to analyze this data in a 4×4 contingency table format here as it was given at DASL. Please make sure to see my book for how this data can be analyzed as two separate explanatory variables through using regression models.

I read in the data from an Excel file and then transform it to a contingency table format:

> diet <- read.csv(file = "C:\\data\\Fiber.csv")

> head(diet)

fiber bloat count

1 bran high 0

2 gum high 5

3 both high 2

4 none high 0

5 bran medium 1

6 gum medium 3

> # Match order given at DASL

> diet$fiber<-factor(x = diet$fiber, levels = c("none",

"bran", "gum", "both"))

> diet$bloat<-factor(x = diet$bloat, levels = c("none",

"low", "medium", "high"))

> diet.table<-xtabs(formula = count ~ fiber + bloat, data =

diet)

> diet.table

bloat

fiber none low medium high

none 6 4 2 0

bran 7 4 1 0

gum 2 2 3 5

both 2 5 3 2

Pay special attention to my use of the factor() function. While it is not necessarily to use here, I use it to order the rows and columns as shown at DASL.

I use three different functions next to test for independence. In practice, you only need to use one of these!

> ind.test<-chisq.test(x = diet.table, correct = FALSE)

> ind.test

Pearson's Chi-squared test

data: diet.table

X-squared = 16.9427, df = 9, p-value = 0.04962

Warning message:

In chisq.test(diet.table, correct = FALSE) :

Chi-squared approximation may be incorrect

> library(package = vcd)

> assocstats(x = diet.table)

X^2 df P(> X^2)

Likelihood Ratio 18.880 9 0.026230

Pearson 16.943 9 0.049621

Phi-Coefficient : 0.594

Contingency Coeff.: 0.511

Cramer's V : 0.343

> class(diet.table)

[1] "xtabs" "table"

> summary(diet.table)

Call: xtabs(formula = count ~ fiber + bloat, data = diet2)

Number of cases in table: 48

Number of factors: 2

Test for independence of all factors:

Chisq = 16.943, df = 9, p-value = 0.04962

Chi-squared approximation may be incorrect

> qchisq(p = 0.95, df = 9)

[1] 16.91898

In summary,

* X2 = 16.94
* -2log(Λ) = 18.88
* 
* P-value using X2 is P(A > 16.94) = 0.0496 where A ~ 
* P-value using -2log(Λ) is P(A > 18.88) = 0.0262 where A ~ 
* Because the p-value is small, but not extremely so, we would say there is moderate evidence against independence (thus, moderate evidence of dependence).
* Thus, there is moderate evidence that bloating severity is dependent on the fiber source.

Can we trust the  approximation? Below are the expected cell counts:

> ind.test$expected

bloat

fiber none low medium high

none 4.25 3.75 2.25 1.75

bran 4.25 3.75 2.25 1.75

gum 4.25 3.75 2.25 1.75

both 4.25 3.75 2.25 1.75

These only partially satisfy the recommendations given earlier.

Alternatives to using a  approximation

Section 6.2 discusses exact inference methods involving Fisher’s exact test and permutation tests. I am going to focus here on using Monte Carlo simulation. Below is a summary of the process:

1. Simulate a large number of contingency tables of size n assuming the null hypothesis is true (i.e., **set** πij =  for one multinomial or **set** πj|i =  for I multinomials); let B be the number of contingency tables.
2. Calculate the X2 (or -2log(Λ)) statistic for each simulated contingency table; call these values ‘s to differentiate them from the original observed X2 value
3. Plot a histogram  and overlay the 
4. Calculate ; this is the p-value for a hypothesis test

Notes:

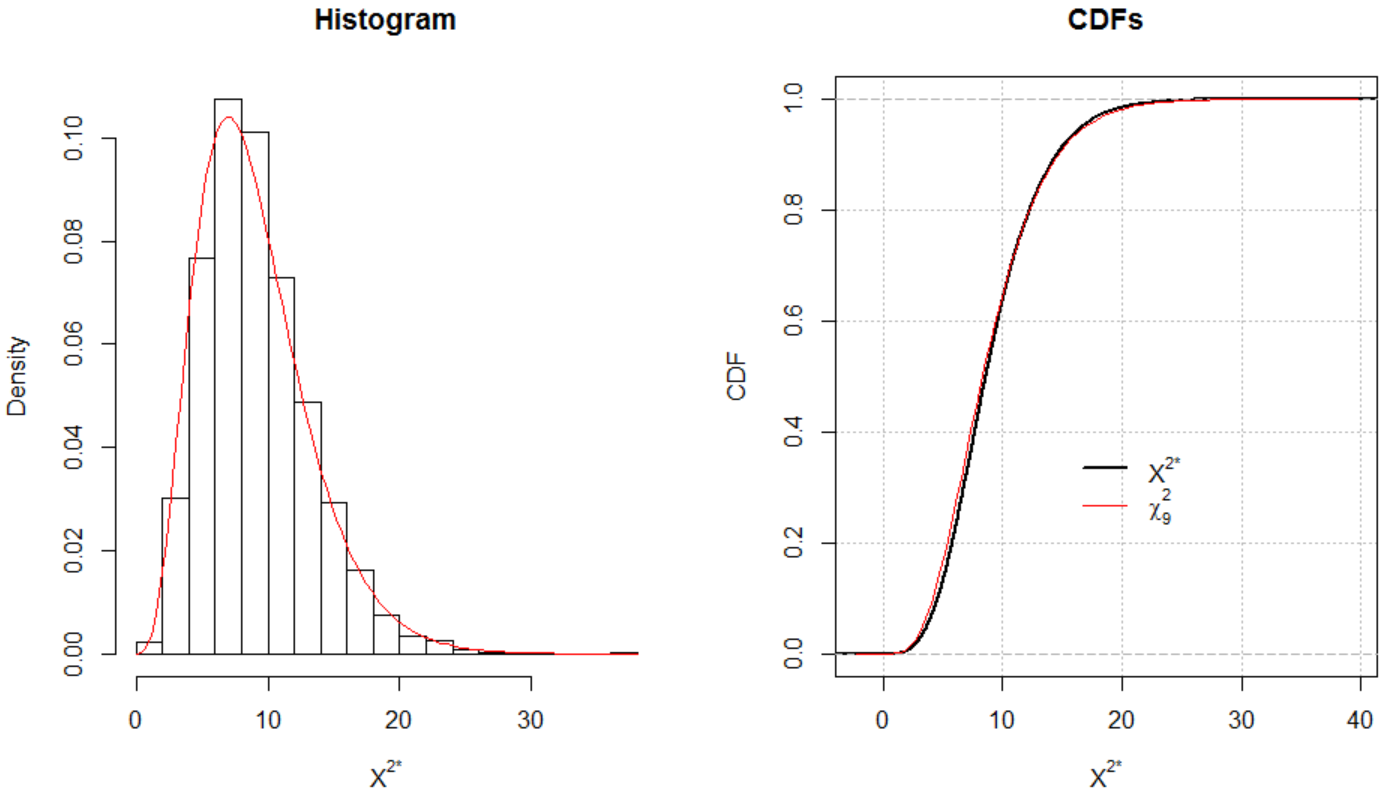
* Step 3 helps to visualize if the original  distribution approximation was appropriate.
* Some simulated contingency tables may have less than I rows or J columns. These contingency tables should be excluded. Why?
* If there are a lot of contingency tables excluded for reasons given in the previous bullet, one should use the methods of Section 6.2 instead.

Example: Fiber enriched crackers (Fiber.R, Fiber.csv)

Please see the homework for the computational details.

Below is a histogram with a  approximation overlaid. Also, I have included a plot of the CDF for a  along with an empirical CDF of the  values.

For a sample of observations , the empirical CDF at w is the proportion of observations at or below w: , where “#” means “number of.”



We see that the chi-square distribution approximation is quite good.

The statistic calculated on the observed data is X2 = 16.94. The p-value using Monte Carlo simulation is 0.04463. When using a  approximation previously, we had a p-value of 0.04962. Through using both inference methods, there is moderate evidence against independence.

If independence is rejected, we would like to determine why it is rejected. For example, perhaps only particular combinations of X and Y are causing the dependence. Also, we would like to determine how much dependence exists. There are a number of ways to examine a contingency table further to understand the dependence. My preference is to generally use statistical models for this purpose, while even using these models to help test for independence. The next two sections describe two types of models that can be used.

**Section 3.3 – Nominal response models**

Suppose there are J categories for the response variable with corresponding probabilities π1, π2, …, πJ. Using the first category as a “baseline”, we can form “baseline category logits” as log(πj/π1) for j = 2, …, J, which are simply log odds.

When J = 2, we have log(π2/π1) = log(π2/(1-π2)), which is equivalent to log(π/(1-π)) in logistic regression with π = π2.

When there is only one explanatory variable x, we can form the multinomial regression model of

log(πj/π1) = βj0 + βj1x for j = 2, …, J

One can easily compare other categories so that category 1 is not always used. For example, suppose you would like to compare category 2 to 3. Then

log(π2/π1) – log(π3/π1) = log(π2) – log(π3) = log(π2/π3)

and

β20 + β21x – β30 – β31x = β20 – β30 + x(β21 – β31)

For more than one explanatory variable, the model becomes:

log(πj/π1) = βj0 + βj1x1 + … + βjpxp for j = 2, …, J

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

We can re-write the model as . Noting that , we have



Thus,



Also, we can now find that



for j = 2, …, J.

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 as given above. Iterative numerical procedures are used then to find the parameter estimates. The multinom() function from the nnet package (within the default installation of R) performs the necessary computations.

The covariance matrix for the parameter estimates follows from using standard likelihood procedures as outlined in Appendix B. Wald and LR-based inference methods are performed in the same ways as for likelihood procedures in earlier chapters.

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

Wheat producers want to identify kernels that are in poor condition after being harvested. To facilitate this identification process, categorization systems have been developed to partition kernels into different categories (see Martin et al. 1998). For this example, we will look at the categories of “Healthy”, “Sprout”, or “Scab”. In summary,

* Healthy is the preferred condition because these kernels have not been damaged
* Sprout is less preferred than healthy because they have reduced weight and poorer flour quality
* Scab is less preferred than healthy because they come from plants that have been infected by a disease and have undesirable qualities in their appearance

Ideally, it would be preferred to make these categorizations for each kernel through using an automated process. To test a new system out, 275 wheat kernels were classified by human examination (assumed to be perfect). The automated system uses information about the class of the wheat kernel (soft red winter or hard red winter) and measurements for density, hardness, size, weight, and moisture for the kernel. Below is part of the data and plots of the data:

> wheat <- read.csv(file = "C:\\data\\wheat.csv")

> head(wheat, n = 3)

class density hardness size weight moisture type

1 hrw 1.349253 60.32952 2.30274 24.6480 12.01538 Healthy

2 hrw 1.287440 56.08972 2.72573 33.2985 12.17396 Healthy

3 hrw 1.233985 43.98743 2.51246 31.7580 11.87949 Healthy

> tail(wheat, n = 3)

class density hardness size weight moisture type

273 srw 0.8491887 34.06615 1.40665 12.0870 11.92744 Scab

274 srw 1.1770230 60.97838 1.05690 9.4800 12.24046 Scab

275 srw 1.0305543 -9.57063 2.05691 23.8185 12.64962 Scab

Below is a parallel coordinate plot (code is in program)



Comments about the parallel coordinate plot:

* Scab kernels generally have smaller density, size, and weight values
* Healthy kernels may have higher densities
* There is much overlap for healthy and sprout kernels
* The moisture content appears to be dependent on hard or soft red winter wheat class

I would like to estimate the following model:

log(πj/π1) = βj0 + βj1x1 + … + βj6x6 for j = 2, 3

What is j = 1, 2, and 3? Again, R uses the same method as we saw in Chapter 2 to order categorical variables.

> levels(wheat$type) #Shows the 3 categories

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

Thus, j = 1 is healthy, j = 2 is scab, and j = 3 is sprout.

Below is how we can estimate a multinomial regression model using the explanatory variables in a linear form:

> library(package = nnet)

> mod.fit<-multinom(formula = type ~ class + density +

hardness + size + weight + moisture, data = wheat)

# weights: 24 (14 variable)

initial value 302.118379

iter 10 value 234.991271

iter 20 value 192.127549

final value 192.112352

converged

> summary(mod.fit)

Call: multinom(formula = type ~ class + density + hardness + size + weight + moisture, data = wheat)

Coefficients:

(Intercept) classsrw density hardness size

Scab 30.54650 -0.6481277 -21.59715 -0.01590741 1.0691139

Sprout 19.16857 -0.2247384 -15.11667 -0.02102047 0.8756135

weight moisture

Scab -0.2896482 0.10956505

Sprout -0.0473169 -0.04299695

Std. Errors:

(Intercept) classsrw density hardness size

Scab 4.289865 0.6630948 3.116174 0.010274587 0.7722862

Sprout 3.767214 0.5009199 2.764306 0.008105748 0.5409317

weight moisture

Scab 0.06170252 0.1548407

Sprout 0.03697493 0.1127188

Residual Deviance: 384.2247

AIC: 412.2247

The estimated models is



and



Notice how R forms an indicator variable for the class of the wheat (“classsrw” corresponds to SRW).

Now that we have the estimated model, many of the basic types of analyses done in Chapter 2 can be performed here. The R code used is very similar as well. Because of the similarity, I would like you to tell me how to do the following:

1. Perform a Wald test for a βjr for some r = 1, …, p.

Why may this test not be of interest?

1. How can we perform a LRT for an explanatory variable?
2. How can we estimate in R the probability of healthy for the first observation?



1. How could we determine a classification for the kernels?
2. What does head(mod.fit$fitted.values) do?

1. How could you estimate the covariance matrix and print it in R?
2. How could you include some type of transformation of an explanatory variable(s) in the model?

Comments:

* The mcprofile package can not be used for likelihood ratio based inference methods.
* Confidence intervals for πj are more complicated to calculate than what we saw in Chapter 2. The main reason is because Brian Ripley, the author of the nnet package, does not believe that one-at-a-time intervals should be calculated. For example, my program shows how to calculate one-at-a-time 95% intervals as

0.7376 < πHealthy < 0.9728

-0.0067 < πScab < 0.0995

0.0143 < πSprout < 0.1825

for the first observation. Of course, πHealthy + πScab + πSprout = 1 needs to occur. If we added the upper limits from the intervals together, we have a total greater than 1! For this reason, Ripley advocates constructing a confidence region. However, this is much more difficult to calculate, and he does not provide any code (no one else provides any code either) to calculate it for these types of models. Still, I think the one-at-a-time intervals provide some useful information, so this is why I give the code in the program to calculate them. Please see my book for a further discussion.

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:



Below is a summary of the estimated probabilities for selected density values:

density.values Healthy Scab Sprout

1 0.8 0.00 0.95 0.05

2 0.9 0.00 0.89 0.11

3 1.0 0.01 0.76 0.24

4 1.1 0.05 0.54 0.41

5 1.2 0.27 0.25 0.48

6 1.3 0.69 0.05 0.25

7 1.4 0.92 0.01 0.07

8 1.5 0.98 0.00 0.02

9 1.6 1.00 0.00 0.00

Notes:

* The lines are drawn from the smallest to the largest observed density value for a wheat kernel condition.
* We see that the estimated scab probability is the largest for the smaller density kernels. The estimated healthy probability is the largest for the high density kernels. For density levels in the middle, sprout has the largest estimated probability. The parallel coordinates plot displays similar findings where the density levels tend to follow the scab < sprout < healthy ordering.

Note that we could also construct a plot like this for more than one explanatory variable, but you would need to condition on values for the other explanatory variables.

**Section 3.3.1 – Odds ratios**

Because the log-odds are being modeled directly in a multinomial regression model, odds ratios are useful for interpreting an explanatory variable's relationship with the response.

Consider the model again of

log(πj/π1) = βj0 + βj1x for j = 2, …, J

The odds of a category j response vs. a category 1 response are exp(βj0 + βj1x). This directly leads to using odds ratios as a way to understand the explanatory variable in the model. Thus, the odds of a category j vs. a category 1 response change by



times for every c-unit increase in x. In a similar manner, we could also compare category j to j′(j ≠ j′, j > 1, j′ > 1):



Make sure that you can show that this!

Notes:

* The odds ratio interpretation specifically states “the odds of a category j vs. a category 1” comparison. In the past when Y was a binary response, we said something like “the odds of a success” only because it was assumed that a comparison was being made to the one other response category (failure).
* When there is more than one explanatory variable, we will need to include a statement like “holding the other variables in the model constant”.
* Similar to what we saw in Section 2.2.5, adjustments need to be made to an odds ratio interpretation when interactions or transformations are present in the model.
* Wald and LR-based inference methods for odds ratios are performed in the same ways as for likelihood procedures discussed in earlier chapters.

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

Because most of the explanatory variables are “continuous”, I decided to use a value of c equal to 1 standard deviation. Ideally, it would be best to talk to the subject-matter researcher about possible values for c. Below is how I estimate the odds ratios for each explanatory variable.

> sd.wheat<-apply(X = wheat[,-c(1,7)], MARGIN = 2, FUN =

sd)

> c.value<-c(1, sd.wheat) #class = 1 is first value

> round(c.value,2)

density hardness size weight moisture

1.00 0.13 27.36 0.49 7.92 2.03

> #beta.hat\_jr for r = 1, ..., 6 and j = 2, 3

> beta.hat2<-coefficients(mod.fit)[1,2:7]

> beta.hat3<-coefficients(mod.fit)[2,2:7]

> #OR for j = 2 (scab vs. healthy)

> round(exp(c.value\*beta.hat2), 2)

density hardness size weight moisture

0.52 0.06 0.65 1.69 0.10 1.25

> round(1/exp(c.value\*beta.hat2), 2)

density hardness size weight moisture

1.91 17.04 1.55 0.59 9.90 0.80

> #OR for j = 3 (sprout vs. healthy)

> round(exp(c.value\*beta.hat3), 2)

density hardness size weight moisture

0.80 0.14 0.56 1.54 0.69 0.92

> round(1/exp(c.value\*beta.hat3), 2)

density hardness size weight moisture

1.25 7.28 1.78 0.65 1.45 1.09

Example interpretations include:

* The estimated odds of a scab vs. a healthy response change by 0.06 times for a 0.13 increase in the density holding the other variables constant. Equivalently, we can say that the estimated odds of a scab vs. a healthy response change by 17.04 times for a 0.13 decrease in the density holding the other variables constant.
* The estimated odds of a sprout vs. a healthy response change by 7.28 times for a 0.13 decrease in the density holding the other variables constant.
* The estimated odds of a scab vs. healthy response change by 9.90 times for a 7.92 decrease in the weight holding the other variables constant.
* The estimated odds of a sprout vs. healthy response change by 1.45 times for a 7.92 decrease in the weight holding the other variables constant. Note that a Wald test of H0: β35 = 0 vs. Ha: β35 ≠ 0, which uses the parameter needed for this sprout vs. healthy odds ratio, has a p-value of 0.2, so this odds ratio may not be interpreted in actual applications.

Relate these odds ratios again back to what we saw in the parallel coordinates plot.

Question: How could estimated odds ratios for sprout vs. scab be calculated?

Below are the calculations for Wald confidence intervals:

> conf.beta<-confint(object = mod.fit, level = 0.95)

> conf.beta #Results are in a 3D array

, , Scab

2.5 % 97.5 %

(Intercept) 22.14 38.95

classsrw -1.95 0.65

density -27.70 -15.49

hardness -0.04 0.00

size -0.44 2.58

weight -0.41 -0.17

moisture -0.19 0.41

, , Sprout

2.5 % 97.5 %

(Intercept) 11.78 26.55

<OUTPUT EDITED>

> ci.OR2<-exp(c.value\*conf.beta[2:7,1:2,1])

> ci.OR3<-exp(c.value\*conf.beta[2:7,1:2,2])

> round(data.frame(low = ci.OR2[,1], up = ci.OR2[,2]), 2)

low up

classsrw 0.14 1.92

density 0.03 0.13

hardness 0.37 1.12

size 0.80 3.55

weight 0.04 0.26

moisture 0.67 2.32

> round(data.frame(low = 1/ci.OR2[,2], up = 1/ci.OR2[,1]),

2)[c(2,5),]

low up

density 7.64 38.00

weight 3.80 25.79

> round(data.frame(low = ci.OR3[,1], up = ci.OR3[,2]), 2)

low up

classsrw 0.30 2.13

density 0.07 0.28

hardness 0.36 0.87

size 0.91 2.59

weight 0.39 1.22

moisture 0.58 1.44

> round(data.frame(low = 1/ci.OR3[,2], up = 1/ci.OR3[,1]),

2)[c(2,3),]

low up

density 3.57 14.82

hardness 1.15 2.74

The density odds ratios can be interpreted as:

With 95% confidence, the odds of a scab instead of a healthy response change by 7.64 to 38.00 times when density is decreased by 0.13 holding the other variables constant. Also, with 95% confidence, the odds of a sprout instead of a healthy response change by 3.57 and 14.82 times when density is decreased by 0.13 holding the other variables constant.

For the scab vs. healthy comparison, only the density and weight odds ratio confidence intervals do not include 1. For the sprout vs. healthy comparison, only the density and hardness odds ratio confidence intervals do not include 1.

Note that there is no function available to automatically calculate profile likelihood ratio intervals.

**Section 3.3.2 – Contingency tables**

The multinomial regression model provides a convenient way to perform the same test for independence as in Section 3.2. We can treat the row variable X as a qualitative variable by constructing I – 1 indicator variables. Using Y as the response variable with category probabilities of π1, …, πJ, we have the model

log(πj/π1) = βj0 + βj2x2 + … + βjIxI for j = 2, …, J

where x2, …, xI are used as indicator variables for X (subscript matches level of X). This is a model under dependence.

A model under independence between X and Y is simply

log(πj/π1) = βj0 for j = 2, …, J

Notice that each category of Y can have a different πj, but they do not change as a function of X.

A test for independence involves the hypotheses of

H0: βj2 = = βjI = 0 for j = 2, …, J

Ha: Not all equal for some j

Equivalently, we can state these hypotheses in terms of models:

H0: log(πj/π1) = βj0 for j = 2, …, J

Ha: log(πj/π1) = βj0 + βj2x2 + … + βjIxI for j = 2, …, J

We can use a LRT to test these hypotheses.

Example: Fiber enriched crackers (Fiber.R, Fiber.csv)

Using bloating severity as the response variable and fiber source as the explanatory variable, a multinomial regression is



where bran, gum, and both in the model represent corresponding indicator variables and the j subscript represents categories low, medium, and high. We can estimate this model using multinom():

> library(package = nnet)

> mod.fit.nom<-multinom(formula = bloat ~ fiber, weights =

count, data = diet2)

# weights: 20 (12 variable)

initial value 66.542129

iter 10 value 54.519963

iter 20 value 54.197000

final value 54.195737

converged

> summary(mod.fit.nom)

Call: multinom(formula = bloat ~ fiber, data = diet2, weights = count)

Coefficients:

(Intercept) fiberbran fibergum fiberboth

low -0.4057626 -0.1538545 0.4055575 1.322135

medium -1.0980713 -0.8481379 1.5032639 1.503764

high -12.4401085 -4.1103893 13.3561038 12.440403

Std. Errors:

(Intercept) fiberbran fibergum fiberboth

low 0.6455526 0.8997698 1.190217 1.056797

medium 0.8163281 1.3451836 1.224593 1.224649

high 205.2385583 1497.8087307 205.240263 205.240994

Residual Deviance: 108.3915

AIC: 132.3915

The weights = count argument in multinom() is used because each row of diet2 represents contingency table counts rather than individual observations.

To perform a LRT for independence, we can use the Anova() function from the car package:

> library(package = car)

> Anova(mod.fit.nom)

# weights: 8 (3 variable)

initial value 66.542129

final value 63.635876

converged

Analysis of Deviance Table (Type II tests)

Response: bloat

LR Chisq Df Pr(>Chisq)

fiber 18.9 9 0.026 \*

---

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

Note that we could have also used the anova() function in the appropriate manner.

The transformed LR statistic value is -2log(Λ) = 18.9, and the p-value is 0.026. These values match what was found earlier using the assocstats() function!

Comments:

* To examine the potential dependence further, we can examine odds ratios in a similar manner to what we did in the wheat example. Please see my book for further details.
* The 0 cell counts are causing the large standard errors for high bloating severity. In fact, a more stringent convergence criteria (use a different value for reltol – see help for the function), will lead to changes in the regression parameter estimates and standard errors. Therefore, we have non-convergence! Fortunately, the only part of the model affected by the non-convergence corresponds to the high bloating severity. Also, the LRT is not affected. Please see my book for a further discussion and an ad-hoc solution to the problem.

When additional categorical explanatory variables are available, we can examine the data in higher dimensional contingency tables through a multinomial regression model. For example, a model for three categorical variables X, Y, and Z can be written as



In this setting, we can examine if X is independent of Y and/or Z is independent of Y in a very similar manner as before. We can also examine if there is a need for an interaction between X and Z.

**Section 3.4 – Ordinal response models**

Suppose that 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 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 β parameter, which means these 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.
* A proportional odds model actually is a special case of a cumulative probability model, which allows the 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) = . 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 are exactly the same, which is a result of β1 being shared for each response category j, except for a horizontal shift, which 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) performs the necessary computations.

It is especially important to have the levels of the categorical response ordered in the desired way when using polr(); otherwise, the ordering of the levels of 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. Wald and LR-based inference methods are performed in the same ways as for likelihood procedures in earlier chapters. 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 on p. 3.52.

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.

While the multinomial regression model result is not necessarily undesirable, it shows that more specific conclusions can be reached for ordinal multinomial responses when the proportional odds model is used.

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$type.order<-factor(wheat$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

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 the estimated parameter 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. Note that the mcprofile package can not be used for likelihood ratio based inference methods with model fits from polr().

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, Ripley does not provide any functions to calculate them. Please see my book and program for one-at-a-time interval code.

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.

**Section 3.4.1 – Odds ratios**

Odds ratios are easily formed because the proportional odds model equates log-odds to the linear predictor. The main difference now is the odds involve cumulative probabilities.

Consider the model again of



The odds ratio is



where  denotes the odds of observing category j or smaller for Y. The formal interpretation of the odds ratio is

The odds of Y ≤ j vs. Y > j change by exp(β1) times for a c-unit increase in x.

Interestingly, this odds ratio stays the same no matter what response category is used for j. This is acgain due the absence of a j subscript on β1 in the model.

Notes:

* When there is more than one explanatory variable, we will need to include a statement like “holding the other variables in the model constant”.
* Similar to what we saw in Section 2.2.5, adjustments need to be made to an odds ratio interpretation when interactions or transformations are present in the model.
* Wald and LR-based inference methods for odds ratios are performed in the same ways as for likelihood procedures discussed in earlier chapters.

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

The estimated odds ratios for each explanatory variable are calculated as  for r = 1, …, 6. Similarly to Section 3.3, c is set to be equal to one standard deviation for each continuous explanatory variable and c = 1 for the SRW variable. Below are the calculations (remember that  is ):

> round(sd(wheat[,-c(1,7)]), 2)

density hardness size weight moisture

0.13 27.36 0.49 7.92 2.03

> c.value<-c(1, sd(wheat[,-c(1,7)])) #class=1

> round(exp(c.value \* (-mod.fit.ord$coefficients)), 2)

density hardness size weight moisture

0.84 0.17 0.75 1.15 0.37 1.08

> round(1/exp(c.value \* (-mod.fit.ord$coefficients)), 2)

density hardness size weight moisture

1.19 5.89 1.33 0.87 2.74 0.92

Example interpretations include:

* The estimated odds of a scab (Y ≤ 1) vs. sprout or healthy (Y > 1) response are 0.84 times as large for soft rather than hard red winter wheat. Note that the corresponding 95% confidence interval for the class variable contains 1 as we will see shortly.
* The estimated odds of a scab vs. sprout or healthy response change by 5.89 times for a 0.13 decrease in the density holding the other variables constant.
* The estimated odds of a scab vs. sprout or healthy response change by 2.74 times for a 7.92 decrease in the weight holding the other variables constant.

Because of the proportional odds, each of the previous interpretations can start with “The estimated odds of a scab or sprout vs. healthy response are ... ,” and the same estimated odds ratios would be used in the interpretation. Also, one could put the interpretation in the following form:

The estimated odds of kernel quality being below a particular level change by \_\_\_ times for a \_\_\_ increase in \_\_\_, holding the other variables constant.

due to the proportional odds.

Overall, we see that the larger the density and weight, the more likely a kernel is healthy. We can again relate these results back to parallel coordinates plot to see why these interpretations make sense.

Below are the calculations for the profile likelihood ratio confidence intervals:

> conf.beta<-confint(object = mod.fit.ord, level = 0.95)

Waiting for profiling to be done...

Re-fitting to get Hessian

> ci<-exp(c.value\*(-conf.beta))

> round(data.frame(low = ci[,2], up = ci[,1]), 2)

low up

classsrw 0.39 1.81

density 0.11 0.26

hardness 0.55 1.03

size 0.77 1.72

weight 0.23 0.58

moisture 0.76 1.54

> round(data.frame(low = 1/ci[,1], up = 1/ci[,2]), 2)

low up

classsrw 0.55 2.57

density 3.87 9.36

hardness 0.97 1.83

size 0.58 1.29

weight 1.73 4.40

moisture 0.65 1.31

Note that the c.value\*(-conf.beta) segment of code multiplies each row of conf.beta by its corresponding element in c.value.

The density odds ratio can be interpreted as:

With 95% confidence, the odds of a scab instead of a sprout or healthy response change by 3.87 to 9.36 times when density is decreased by 0.13 holding the other variables constant.

Similar modifications to the above interpretation can be made as before due to the proportional odds assumption.

Wald confidence intervals may be calculated as well, but the confint.default() function can not be applied. Please see the code in the program for how to perform these calculations without using this function.

**Section 3.4.2 – Contingency tables**

A proportional odds model can not be used to perform the exact same type of test for independence that we saw in Sections 3.2 and 3.3. As mentioned earlier in this sub-section, the alternative hypothesis with this model specifies one form of the dependence by taking into account the ordinal response. This can be advantageous because a smaller alternative hypothesis, leads to a more powerful test.

Suppose we have a categorical variable X that is represented in our model by I – 1 indicator variables. Also, suppose we have a ordinal categorical response Y with category probabilities of π1, …, πJ. A model under independence between X and Y is simply



for j = 1, …, J – 1. A model allowing for dependence is



where x2, …, xI are used as indicator variables for X (subscript matches level of X). A test of independence involves the hypotheses:

H0:  for j=1,…,J–1

Ha:  for j=1,…,J–1

Notice the alternative hypothesis is not as general as what we had in Section 3.3.2 with the multinomial regression model:

H0: log(πj/π1) = βj0 for j = 2, …, J

Ha: log(πj/π1) = βj0 + βj2x2 + … + βjIxI for j = 2, …, J

which did not say what type of dependence existed.

Example: Fiber enriched crackers (Fiber.R, Fiber.csv)

We previously found that there was *marginal* evidence of dependence between bloating severity and fiber source. Because bloating severity is measured in an ordinal manner (none < low < medium < high), a proportional odds model allows us to take this into account more easily and perhaps reach stronger conclusions about the problem of interest.

The alternative hypothesis model here is



where j corresponds to levels 1 (none), 2 (low), 3 (medium), and 4 (high) of bloating severity.

Below is how I estimate the model:

> library(package = MASS)

> levels(diet$bloat)

[1] "none" "low" "medium" "high"

> mod.fit.ord<-polr(formula = bloat ~ fiber, weights =

count, data=diet2, method = "logistic")

> summary(mod.fit.ord)

Re-fitting to get Hessian

Call: polr(formula = bloat ~ fiber, data = diet2, weights = count, method = "logistic")

Coefficients:

Value Std. Error t value

fiberbran -0.3859 0.7813 -0.494

fibergum 2.4426 0.8433 2.896

fiberboth 1.4235 0.7687 1.852

Intercepts:

Value Std. Error t value

none|low 0.0218 0.5522 0.0395

low|medium 1.6573 0.6138 2.7002

medium|high 3.0113 0.7249 4.1539

Residual Deviance: 112.2242

AIC: 124.2242

The estimated model is



where . A LRT for the fiber source variable gives:

> library(package = car)

> Anova(mod.fit.ord)

Analysis of Deviance Table (Type II tests)

Response: bloat

LR Chisq Df Pr(>Chisq)

fiber 15.048 3 0.001776 \*\*

Because  = 15.048 is large relative to a distribution (p-value = 0.0018), there is strong evidence that a dependence exists in the form of a trend among the log-odds for the cumulative probabilities. Remember again that the LRT for independence with a more general alternative hypothesis only concluded marginal evidence of dependence and did not specify what type of dependence.

This dependence can be further investigated through examining the odds ratios. Please see the book for a discussion. In the end, there is sufficient evidence to indicate a trend among bloating severity for the gum fiber source.

**Section 3.4.3 – Non-proportional odds model**

A proportional odds model is one of the preferred ways to account for an ordered multinomial response, because it includes only one parameter for each explanatory variable. While this can greatly simplify the model, it may not work well for some problems.

A more general model is

 for j = 1, …, J - 1

which I will simply refer to as a non-proportional odds model (a another type of cumulative probability model). Notice the j subscript on the βj1, …, βjp parameters in the model. By allowing for the β’s to change for each response category, we can construct a test of the proportional odds assumption:

H0: β1r = = βJ-1,r for r = 1, …, p

Ha: Not all equal for some r

If the proportional odds assumption is rejected, it may be preferred to use the non-proportional odds model for the data analysis of interest. However, the proportional odds model may still be preferred due to its smaller number of parameters. For example, a very large sample size could result in a rejection of the null hypothesis even though there is little practical violation of the assumption.

There is another reason why the proportional odds model may be preferred. The non-proportional odds model allows for

P(Y ≤ j) < P(Y ≤ j′) for j > j′

Why is this a problem?

For this reason, care needs to be used with these models so that nonsensical probabilities do not occur.

Example: Non-proportional odds model plots (CumulativeLogitModelPlot.R)

The purpose here is to show the problems that can occur with the non-proportional odds model. Consider the model of



where  and J = 4. Notice this is the same model as we saw at the beginning of this section except  and 

Below is a plot of the model:



Notice that the cumulative probability curve for P(Y ≤ 3) crosses the other two curves! This means that P(Y ≤ 3) < P(Y ≤ 2) and P(Y ≤ 3) < P(Y ≤ 1) for some value of x1. As a result, π3 < 0 for some x1, which can be seen in the plot of the right.

The LRT for the proportional odds assumption can not be performed through using the polr() function. Instead, I show in my book how to use the vglm() function from the VGAM package to help perform the test.

**Section 3.5 – Additional regression models**

There are a number of other models that can be used with multinomial responses. One of these is the adjacent-categories model:



where j = 1, …, J – 1, which is somewhat similar to the multinomial regression model from Section 3.3. To take advantage of an ordinal response, one could also use the model



where j = 1, …, J – 1. Notice the removal of the j subscript on the last p – 1 β parameters. Both of these models can be fit in R using the vlgm() function of the VGAM package.