Multinomial Regression

Adam J Sullivan, PhD
04/02/2018

Multinomial Regression

Multinomial

Many times our data is not just in a success or failure format. For example in the Physicians' Health Study the primary cardiovascular endpoint was:

- : No Cardiovascular Disease
- : Myocardial Infarction
- : Stroke
- : Other CV Death

What Is Different than Logistic?

- We may not want to just focus on successes and failures but all of the categories.
- Instead of a binomial distribution like in logistic regression we have a multinomial distribution.

The Multinomial Distribution

- The is a generalization of the binomial distribution.
  - The binomial distribution had n independent trials and each trial had 2 outcomes, `Success` and `Failure`.
  - In the Multinomial there are r outcomes for each of the n trials. and each outcome has probability $p_1, \ldots, p_r$ of occurring $(p_1 + \cdots p_r = 1)$.

The Multinomial Distribution

- Let $ N_i$ denote a count of the number of type i outcomes in the n trials, $i = 1, 2, \ldots, r$.
- For the joint function we are concerned with a sequence of trials that gives us $N_1 = n_1, \ldots, n_r = n_r$ which occurs with probability $p_1^{n_1} \cdots p_r^{n_r}$.
- The number of ways to order n outcomes is

$$\binom{n}{n_1 \cdots n_r} = \frac{n!}{n_1! \cdots n_r!}$$

thus the joint PMF is

$$p(n_1, \ldots, n_r) = \binom{n}{n_1 \cdots n_r} p_1^{n_1} \cdots p_r^{n_r}$$

Further About the Categories

- This looks very much like the binomial distribution with the exception that there are more categories a subject can be in.
- We have two situations that can come from this data and that informs how we analyze it. We can have
  - Nominal Categories
  - Ordinal Categories

Types of Regressions

- With these types of categories we will proceed with 2 types of multinomial regressions:
  - Baseline Category Logit
  - Adjacent Category Logit
  - Proportional-Odds Cumulative Logit

Baseline Category Logit

- If you recall in logistic regression we have 2 probabilities:
  - π_1: Probability of having a success.
  - π_0: Probability of having a failure.

Baseline Category Logit

- We then run our logistic regression as

$$\log\left(\frac{\pi_1}{\pi_0}\right) = logit(\pi_1) = \beta_0 + \beta_1 x_1 + \cdots + \beta_k x_k$$

- With baseline category logit we have multiple probabilities:
  - π_1 : Probability of being in category 1.
  - π_2 : Probability of being in category 2.
  - $\vdots$
  - π_J : probability of being in category J.

How Do we Do It?

We first pick category J to be our baseline, we then proceed to fit $J-1$ logistic regressions simultaneously:

$$\begin{aligned}
\log\left(\frac{\pi_1}{\pi_J}\right) &= \alpha_1 + \beta_{1,1}x_1 + \cdots + \beta_{k,1}x_k \\
\log\left(\frac{\pi_2}{\pi_J}\right) &= \alpha_2 + \beta_{1,2}x_1 + \cdots + \beta_{k,2}x_k \\
&\vdots \\
\log\left(\frac{\pi_{J-1}}{\pi_J}\right) &= \alpha_{J-1} + \beta_{1,J-1}x_1 + \cdots + \beta_{k,J-1}x_k
\end{aligned}$$

Interpreting?

- We then can interpret the odds ratios exactly as we have before.
- The only difference is that this time it is not odds of having outcome vs not having the outcome but being in a particular category vs category J.

Comparisons of Any Pair of Categories

- We can then take any 2 categories, say a and b and find a logit function to compare them:

$$\begin{aligned}\log\left(\frac{\pi_a}{\pi_b}\right) &= \log\left(\frac{\frac{\pi_a}{\pi_J}}{\frac{\pi_b}{\pi_J}}\right)\\ &= \log\left(\frac{\pi_a}{\pi_J}\right) - \log\left(\frac{\pi_b}{\pi_J}\right)\\ &= (\alpha_a - \alpha_b) + (\beta_{1,a} - \beta_{1,b})\, x_1 + \cdots + (\beta_{k,a} - \beta_{k,b})\, x_k\end{aligned}$$

An Example

- We begin with an example from The General Social Surveys (1991).
- Among the questions asked to American citizens were questions regarding beliefs in an after life.
- The table below displays the data for this:

		BELIEF IN AFTERLIFE		
RACE	SEX	YES	UNDECIDED	NO
White	Female	372	49	74
	Male	250	45	71
Black	Female	64	9	15
	Male	252	5	13

We can read this data into R:

```
after.life <- read.table("after_life.csv", header=TRUE, sep=",")
after.life$race <- factor(after.life$race, levels = c(0,1), labels=c("Black", "White"))
after.life$sex <- factor(after.life$sex, levels = c(0,1), labels=c("Male", "Female"))
after.life$belief <- factor(after.life$belief, levels = c(2,1,3),
                            labels=c("Undecided","Yes",  "No"))
```

Our Model

Then we can fit a baseline category logit model:

```
library(VGAM)
fit1 <- vglm(belief ~ race + sex, data=after.life, weights=count,
             family=multinomial)
```

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
(Intercept):1	-0.714	0.49	(0.24 , 0.999)	0.0498
(Intercept):2	0.926	2.524	(1.559 , 4.087)	0.0002
raceWhite:1	0.234	1.263	(0.629 , 2.538)	0.5118
raceWhite:2	0.306	1.358	(0.849 , 2.173)	0.2022
sexFemale:1	0.091	1.096	(0.675 , 1.778)	0.7109
sexFemale:2	0.405	1.5	(1.071 , 2.099)	0.0183

What Do We See?

- We can see that very little is significant here.
- However we will interpret one value that is significant: `sexFemale:2`, We can use the table for the odds ratio and find that for people of the same race females have 50% higher odds of "undecided" vs "no" on after life beliefs than males do.

Adjacent Category Logit

- When we have ordinal data we can use different methods than before.
- One of those methods is called adjacent category logits.
- In this case we can assume that our outcome has r levels which are ordered.

Adjacent Category Logit

Then we set out to compare the probabilities of

- outcome 1 vs outcome 2
- outcome 2 vs outcome 3
- outcome 3 vs outcome 4
- $\vdots$
- outcome $r-1$ vs outcome r

Our Logits

- This would then lead to the following logits:

$$
\begin{aligned}
L_1 &= \log\left(\frac{\pi_1}{\pi_2}\right) = \alpha_1 + \beta_{1,1}x_1 + \cdots + \beta_{k,1}x_k \\
L_2 &= \log\left(\frac{\pi_2}{\pi_3}\right) = \alpha_2 + \beta_{1,2}x_1 + \cdots + \beta_{k,2}x_k \\
&\vdots \\
L_{r-1} &= \log\left(\frac{\pi_{r-1}}{\pi_r}\right) = \alpha_{r-1} + \beta_{1,r-1}x_1 + \cdots + \beta_{k,r-1}x_k
\end{aligned}
$$

Out Models

We can also use these models to create the baseline category logit model with say reference as category 1:

$$\begin{aligned}
\log\left(\frac{\pi_2}{\pi_1}\right) &= -L_1 \\
\log\left(\frac{\pi_3}{\pi_1}\right) &= -L_2 - L_1 \\
\log\left(\frac{\pi_4}{\pi_1}\right) &= -L_3 - L_2 - L_1 \\
&\vdots \\
\log\left(\frac{\pi_r}{\pi_1}\right) &= -L_{r-1} - \cdots - L_3 - L_2 - L_1
\end{aligned}$$

Recall Self Assessed Health

VARIABLE	DESCRIPTION
id	Identification Number
age	Age (years)
male	Sex
	1 = Male
	0 = Female
pperf	Physical Performance Scale (0-12)
pefr	Peak Expiratory Flow Rate (average of 3)
cogerr	Number of cognitive errors on SPMSQ

VARIABLE	DESCRIPTION
sbp	Systolic Blood Pressure (mmHg)
mile	Ability to Walk 1 mile
	0 = No
	1 = Yes
digit	Use of digitalis
	0 = No
	1 = Yes

VARIABLE	DESCRIPTION
loop	Use of Loop Diuretics
	0 = No
	1 = Yes
untrt	Diagnosed but untreated diabetes
	0 = No
	1 = Yes
trtdb	Treated Diabetes
	0 = No
	1 = Yes

VARIABLE	DESCRIPTION
sah	Self Assessed Health
	1 = Excellent
	2 = Good
	3 = Fair
	4= Poor
dead	Dead by 1991
	0 = No
	1 = Yes
time	Follow up (years)

```r
library(haven)
sah <- read_dta("sah.dta")
str(sah$sah)
#Reverse order for self assessed health
sah$sah2 <- factor(sah$sah, labels = names(attributes(sah$sah)$labels))
sah$sah <- factor(sah$sah2, levels = rev(levels(sah$sah2)))
```

```
## Class 'labelled'  atomic [1:1600] 2 2 2 4 3 1 2 3 1 2 ...
##   ..- attr(*, "label")= chr "Self assessed health"
##   ..- attr(*, "format.stata")= chr "%8.0g"
##   ..- attr(*, "labels")= Named num [1:4] 1 2 3 4
##   .. ..- attr(*, "names")= chr [1:4] "excellent" "good" "fair" "poor"
```

Our Model

- Before we were interested in modeling `death` this time we will attempt to model `sah`.
- We will attempt to build a model with the following covariates:
  - `pperf`
  - `perf`
  - `cogerr`
  - `mile`
  - `loop`
  - `age`

Running the Model

```
library(VGAM)
fit2 <- vglm(as.ordered(sah) ~ pperf + perf + cogerr + mile + loop + age, data=sah,
             family=acat(reverse=TRUE))
```

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
(Intercept):1	-1.441	0.237	(0.008 , 7.259)	0.4094
(Intercept):2	4.659	105.534	(12.025 , 926.218)	0
(Intercept):3	5.171	176.121	(8.224 , 3771.667)	0.0009
pperf:1	-0.203	0.816	(0.748 , 0.891)	0
pperf:2	-0.117	0.89	(0.85 , 0.931)	0
pperf:3	-0.069	0.933	(0.872 , 0.998)	0.0436

More of the Table

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
perf:1	-0.001	0.999	(0.996 , 1.001)	0.2397
perf:2	0	1	(0.998 , 1.001)	0.4003
perf:3	-0.001	0.999	(0.998 , 1.001)	0.3527
cogerr:1	0.004	1.004	(0.89 , 1.134)	0.9424
cogerr:2	0.01	1.01	(0.935 , 1.09)	0.8062
cogerr:3	-0.05	0.952	(0.852 , 1.063)	0.3797
mile:1	-0.311	0.733	(0.418 , 1.283)	0.2766
mile:2	-0.863	0.422	(0.317 , 0.561)	0
mile:3	-0.579	0.56	(0.337 , 0.932)	0.0256

More of the Table

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
mile:3	-0.579	0.56	(0.337 , 0.932)	0.0256
loop:1	0.187	1.206	(0.686 , 2.117)	0.5155
loop:2	0.333	1.395	(0.92 , 2.115)	0.1166
loop:3	-0.448	0.639	(0.337 , 1.21)	0.1693
age:1	0.015	1.015	(0.974 , 1.057)	0.4818
age:2	-0.046	0.955	(0.931 , 0.98)	0.0005
age:3	-0.031	0.969	(0.935 , 1.004)	0.0849

Interpreting

- We will not spend time interpreting this as the next method is more used than adjacent category logits.

Proportional-Odds Cumulative Logit

- Proportional-Odds Cumulative Logit is the most popular for ordinal categorical outcome analysis.
- It consists of probabilities
  - π_1: Probability of being in category 1.
  - π_2: Probability of being in category 2.
  - $\vdots$
  - π_J: probability of being in category J.

Our Probabilities

- We also consider 2 more probabilities
  - $\Pr(Y \leq j) = \pi_1 + \cdots + \pi_j$
  - $\Pr(Y > j) = \pi_{j+1} + \cdots + \pi_J$
- We define the cumulative logit as

$$\log\left(\frac{\Pr(Y \leq j)}{\Pr(Y > j)}\right) = \log\left(\frac{\Pr(Y \leq j)}{1 - \Pr(Y \leq j)}\right) = \log\left(\frac{\pi_1 + \cdots + \pi_j}{\pi_{j+1} + \cdots + \pi_J}\right)$$

Cumulative Logits

- We then have the following cumulative logits

$$
\begin{aligned}
L_1 &= \log\left(\frac{\pi_1}{\pi_2 + \pi_3 + \cdots + \pi_J}\right) \\
L_2 &= \log\left(\frac{\pi_1 + \pi_2}{\pi_3 + \pi_4 \cdots + \pi_J}\right) \\
&\vdots \\
L_{J-1} &= \log\left(\frac{\pi_1 + \pi_2 + \cdots + \pi_{J-1}}{\pi_J}\right)
\end{aligned}
$$

An Example

We can continue on with out Self Assessed Health and run the following model:

```
library(VGAM)
fit3 <- vglm(as.ordered(sah) ~ pperf + perf + cogerr + mile + loop + age, data=sah,
             family=cumulative)
```

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
(Intercept):1	-0.238	0.788	(0.034 , 18.471)	0.8822
(Intercept):2	4.74	114.488	(15.714 , 834.114)	0
(Intercept):3	7.628	2054.927	(109.59 , 38531.939)	0
pperf:1	-0.256	0.774	(0.712 , 0.841)	0
pperf:2	-0.154	0.857	(0.822 , 0.894)	0
pperf:3	-0.126	0.882	(0.827 , 0.941)	0.0001

More of the Table

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
perf:1	-0.002	0.998	(0.995 , 1)	0.0602
perf:2	-0.001	0.999	(0.998 , 1)	0.2011
perf:3	-0.001	0.999	(0.998 , 1.001)	0.3289
cogerr:1	0.035	1.035	(0.926 , 1.157)	0.5403
cogerr:2	0.003	1.003	(0.935 , 1.077)	0.9235
cogerr:3	-0.05	0.951	(0.856 , 1.057)	0.3508
mile:1	-0.868	0.42	(0.245 , 0.72)	0.0016
mile:2	-0.994	0.37	(0.284 , 0.482)	0
mile:3	-1.051	0.35	(0.213 , 0.575)	0

More of the Table

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
mile:3	-1.051	0.35	(0.213 , 0.575)	0
loop:1	0.236	1.267	(0.74 , 2.17)	0.3891
loop:2	0.268	1.307	(0.899 , 1.901)	0.1612
loop:3	-0.237	0.789	(0.43 , 1.448)	0.4443
age:1	-0.003	0.997	(0.96 , 1.035)	0.8612
age:2	-0.042	0.959	(0.936 , 0.981)	0.0004
age:3	-0.046	0.955	(0.923 , 0.988)	0.0087

What Can We See?

- Before we move on, we notice that our coefficients do not seem to be that different between logits.
- We can change this so that instead of having a coefficient of `pperf` for each cumulative logit we can have one coefficient that is the same for all of the models giving each of them different intercepts only:

The Model

```
library(VGAM)
fit4 <- vglm(as.ordered(sah) ~ pperf + perf + cogerr + mile + loop + age, data=sah,
             family=cumulative(parallel = TRUE))
```

	ESTIMATE	ODDS RATIO	95% CI OF OR	P-VALUE
(Intercept):1	1.837	6.279	(1.121 , 35.188)	0.0367
(Intercept):2	4.399	81.391	(14.396 , 460.16)	0
(Intercept):3	7.097	1208.849	(209.132 , 6987.522)	0
pperf	-0.163	0.85	(0.818 , 0.882)	0
perf	-0.001	0.999	(0.998 , 1)	0.115
cogerr	-0.005	0.995	(0.935 , 1.059)	0.8684
mile	-1.006	0.366	(0.286 , 0.468)	0
loop	0.189	1.208	(0.865 , 1.687)	0.2672
age	-0.036	0.964	(0.945 , 0.984)	0.0005

Interpretations

- Then we can interpret `mile` as:
    -

- The same things can be done to our other two models by adding the `parallel=TRUE` option in the family statement.