MATH411 | Fall 2018 | Chapter 7: Discrete Response Models

Dr. Yongtao Cao

November 26, 2018

Contents

7.1 Introduction

When the outcome (response) variable is expressed as a continuous variable, a linear regression model is often adequate to describe the impact of other factors on this variable, or to predict the response value(s) in altered circumstances. In many applications, however, the researcher is often faced with a discrete rather than a continuous outcome variable. In general, we focus on models where the outcome variable takes the values of $0, 1, 2, \ldots$ (**Note**: these are not always numbers, as you can see below, they can be labels).

Case I: Binary Response: $y_i \in \{0, 1\}$

- Patient survival: `yes` or `no`
- Customer retention: `churn` or `not`

Case II: More than 2 Alternatives: $y_i \in \{0, 1, 2, \ldots, J\}$

There are also two cases:

1. Values of y are only indicators for different alternatives, order is irrelevant → **multinomial response model**

- Brands of consumption goods
- Choice of holiday destination

2. Values of y indicate order of alternatives → **ordered response model**

- Ratings
- Development of profits
- Satisfaction with life / income

Case III: Response Outcomes are Discrete Counts: $y_i \in \{0, 1, 2, \ldots\}$

- Goals scored by Pittsburgh Penguins during a game

7.2 Binary Classification

7.2.1 Bernoulli and Binomial Probability Distributions

Suppose $Y = 1$ is a success where the probability of a **success** is $P\left(Y = 1\right) = \pi$, $Y = 0$ is a **failure**.

Bernoulli Probability Mass Function:

$$P\left(Y = y\right) = \pi^{y}\left(1 - \pi\right)^{1-y} \;\; for \; y = 0, \; or \; 1$$

Often, you observe multiple success/failure observations. Let $Y_1, \ldots, Y_n$ denote random variables for these observations. If the random variables are **independent and have the same probability of success** π, then we can use a **binomial** PMF for $W = \Sigma_{i=1}^{n} Y_i$.

Binomial Probability Distributions

$$P\left(W=w\right)=\frac{n!}{w!\left(n-w\right)!}\pi^{w}\left(1-\pi\right)^{n-w}\;\; for\; w=0,1,\ldots,n$$

Required Conditions When Applying Binomial Probability Model

1. There are n identical trials.
2. Each trial has two possible outcomes, typically referred to as a success or failure.
3. The trials are independent of each other.
4. The probability of success, denoted by π, remains constant for each trial. The probability of a failure is $1-\pi$.

Odds

Odds are the probability of success divided by the probability of a failure: $\frac{\pi}{1-\pi}$.

Odds are a rescaling of the probability of success.

- If $P(success) = 0.75$, then the odds are 3 or "3 to 1 odds," that is, **the probability of a success are three times as large as the probability of a failure**.

Note that:

- Probability ranges from ____________.
- Odds range from ____________.
- Log Odds range from ____________.

7.2.2 Logistic Regression

Response Probability

In a binary response model, we model the response probability

where

- y_i is
- x_{ij} is

By convention, we use π to denote the response probability, the above model can also be written as

To the precise, this is a **conditional response probability**, conditional on the set of predictors $\mathbf{x}_i$ of individual i.

Case Study I: Wolf Habitat Suitability in the Northern Rocky Mountains

The `wolves.csv` data file contains the following variables:

- `gridcell` IDs for the 30×30 grid cells.
- `MAJOR_LC = codes` for major land cover types (descriptions in `landcover.txt`)

- `RD_DENSITY` is a measure of, well, road density.

- `WOLVES_99 = 2` means the data came after 1999.

- `WOLVES_01` The response variable, also will be used to test the prediction ability of the logistic regression model.
- Paper: `Houts03.pdf`

Let's do the math with the `wolf99` data step by step to see the transformation from probability to odds to log odds:

From the R output, we can see that:

- **The probability of wolf presence** $\hat{\pi} =$
- **The odds of the probability of wolf presence** $odds =$
- **The log odds of the probability of wolf occurrence** $log(odds) =$

Logistic Regression with No Predictor Variables

Let's start with a simple model without any predictors:

$$logit(\pi) = log(\frac{\pi}{1-\pi}) = \beta_0$$

or in terms of probabilities:

- The intercept $= -0.4109$ which corresponds to
- We can go from the log odds to the odds by exponentiating the coefficient which gives us the odds:
- We can go backwards to the probability by calculating:
- This model, actually, is nothing but the **one proportion Z test** that you have seen from MATH214/6/7!

Logistic Regression with a Single Continuous Predictor Variable

Let us build a model with `RD_DENSITY` as the predictor. But first, plot `WOLVES_99` vs `RD_DENSITY`.

- The model being fit is:

So our estimated model is given by:

- The intercept $= 0.3261$ which is interpreted as the log odds of wolves presence for a road density of 0.

- The coefficient for road density $= -1.0169$ which is interpreted as the expected change in log odds for a one-unit increase in the road density.

- The odds ratio can be calculated by exponentiating this value to get exp(-1.0169) which means **we expect to see about 36.17% decrease in the odds of wolve presence for a one unit increase in road density**.

From this fitted model:

- The predicted `log odds` of wolf presence for a grid with a road density of 0.9 is:

- The predicted `odds` of wolf presence for a grid with a road density of 0.9 is thus:

- Since the odds are less than 1, we should find that the probability of wolf presence is less than $\frac{1}{2}$. To be specific, we see that the predicted probability of wolf presence for a grid with a road density of 0.9 is

In general, we can find the model in terms of a statement about probabilites, like this

Using `predict` to Describe the Model's Fits

To obtain these fitted odds and probabilities in R, we can use the `predict` function.

- The default predictions are on the scale of the log odds. These predictions are also available through the `type = "link"` command within the `predict` function for a generalized linear model like logistic regression.
- We can also obtain predictions for each subject on the original response (here, `probability`) scale, backing out of the logit link.

Interpreting the Model Summary

Let's get a more detailed summary of our `f1` model, including 95% confidence intervals for the coefficients:

Some elements of this summary are very familiar from our work with linear models.

- We still have a five-number summary of residuals, although these are called *deviance* residuals.
- We have a table of coefficients with standard errors, and hypothesis tests, although these are **Wald z-tests**, rather than the t tests we saw in linear modeling.
- We have a summary of global fit in the comparison of null deviance and residual deviance, but without a formal p value.
- And we have the AIC.
- We also have some new items related to a *dispersion* parameter and to the number of Fisher Scoring Iterations.

Let's walk through each of these elements.

1. **Wald Z tests** for Coefficients in a Logistic Regression

The coefficients output provides the estimated coefficients, and their standard errors, plus a Wald Z statistic, **which is just the estimated coefficient divided by its standard error**. This is compared to a standard Normal distribution to obtain the two-tailed p values summarized in the `Pr(>|z|)` column.

- The hypotheses being tested here are H_0: `RD_DENSITY` does not have an effect on the log odds of `Presence` vs. H_a: `RD_DENSITY` does have such an effect.
- Another way of stating this is that the *p* value assesses whether the estimated coefficient of `RD_DENSITY`, -1.0169, is statistically detectably different from 0. If the coefficient (on the logit scale) for `RD_DENSITY` was truly 0, this would mean that:
  - the log odds of wolf presence did not change based on the `RD_DENSITY` size,
  - the odds of wolf presence were unchanged based on the `RD_DENSITY` size (the odds ratio would be 1), and
  - the probability of wolf presence was unchanged based on the `RD_DENSITY` size.

In our case, we have a statistically detectable change in the log odds of `wolf presence` associated with changes in `RD_DENSITY`, according to this *p* value. We conclude that `RD_DENSITY` size is associated with a negative impact on wolf presence rates (wolf presence rates are generally lower for grid with higher road density.)

2. Confidence Intervals for the Coefficients

As in linear regression, we can obtain 95% confidence intervals (to get other levels, change the `level` parameter in `confint`) for the intercept and slope coefficients.

Here, we see, for example, that the coefficient of `RD_DENSITY` has a point estimate of -1.0169, and a confidence interval of (-1.7139, -0.3988). Since this is on the logit scale, it's not that interpretable, but we will often exponentiate the model and its confidence interval to obtain a more interpretable result on the odds ratio scale.

- From the output, we can estimate the odds ratio for wolf presence associated with a 1 unit increase in road density is 0.3617, with a 95% CI of (0.16, 0.67). If the odds ratio was 1, it would indicate that the odds of wolf presence did not change based on the change in road density.

3. Deviance Residual

In logistic regression, it's certainly a good idea to check to see how well the model fits the data. However, there are a few different types of residuals. The residuals presented here by default are called **deviance residuals**. The deviance residuals for each individual subject sum up to the deviance statistic for the model, and describe the contribution of each point to the model likelihood function.

The deviance residual, d_i, for the i$^{\text{th}}$ observation in a model predicting y_i (a binary variable), with the estimate being $\hat{\pi}_i$ is:

$$d_i = s_i\sqrt{-2[y_i log\hat{\pi}_i + (1-y_i)log(1-\hat{\pi}_i)]},$$

where s_i is 1 if $y_i = 1$ and $s_i = -1$ if $y_i = 0$. Again, the sum of the deviance residuals is the deviance.

- The deviance is available for both the null model (the model with only an intercept) and for our model (`f1`) predicting our outcome, wolf presence.
- The deviance test, though available in R isn't really a test of whether the model works well. Instead, it assumes the model is true, and then tests to see if the coefficients are detectably different from zero. So it isn't of much practical use.

4. Dispersion Parameter

The dispersion parameter is taken to be 1 for `glm` fit using either the `binomial` or `Poisson` families. For other sorts of generalized linear models, the dispersion parameter will be of some importance in estimating standard errors sensibly.

5. Fisher Scoring iterations

The solution of a logistic regression model involves maximizing a likelihood function. Fisher's scoring algorithm in our `f1` needed three iterations to perform the logistic regression fit. All that tells you is that the model converged, and didn't require a lot of time to do so.

Logistic Regression with a Single Continuous Predictor Variable and a Categorical Predictor

Now let's add `Landtype` as a predictor.

How well does A Model Classify Subjects?

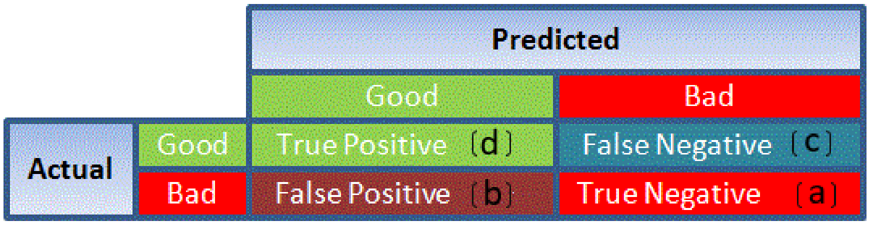
A natural question to ask is how well does our model classify grids in terms of likelihood of wolf habitat suitability.

We could specify a particular rule, for example: **if the predicted probability of wolf presence is 0.5 or greater, then predict "suitable habitat"**.

- The **default cutoff prediction probability score is 0.5** or the ratio of 1's and 0's in the training data. But sometimes, **tuning the probability cutoff can improve the accuracy in both the development and validation samples**.
- Try 0.5 in our case, see R code.

Confusion Matrix

A confusion matrix is nothing but a tabular representation of `Actual` vs `Predicted` values. This helps us to find the accuracy of the model and avoid overfitting. This is how it looks like:



		Predicted	
		Good	Bad
Actual	Good	True Positive (d)	False Negative (c)
	Bad	False Positive (b)	True Negative (a)

Figure 1: Confusion Matrix

From confusion matrix, **Specificity** and **Sensitivity** can be derived as illustrated below:

$$\left.\begin{aligned} \text{True Negative Rate (TNR), specificity} &= \frac{A}{A+B} \\ \text{False Positve Rate (FPR), } 1-\text{specificity} &= \frac{B}{A+B} \end{aligned}\right\}\text{sum to 1}$$

$$\left.\begin{aligned} \text{True Positive Rate (TPR), sensitivity} &= \frac{D}{C+D} \\ \text{False Negative Rate (FNR)} &= \frac{C}{C+D} \end{aligned}\right\}\text{sum to 1}$$

Figure 2: Specificity and Sensitivity

- Let us calculate these quantities from the output.
- **Goal: try to minimize FPR and simultaneously maximize TPR**.

ROC Curve

`Receiving Operating Characteristic`, or `ROC`, is a visual way for inspecting the performance of a binary classifier (0/1). It plots `FPR` on the x-axis and TPR (sensitivity) on the y-axis. The point at $(0, 1)$ indicates perfect predictions, and the diagonal line indicates random guessing (thus curves closer to the upper left corner indicate better models).

ROC curves were first used during WWII to analyze radar effectiveness. In the early days of radar, it was sometimes hard to tell a bird from a plane. The British pioneered using ROC curves to optimize the way that they relied on radar for detecting incoming German planes.

Figure 3: Blast, birds again!

Let's go over a few simple rules for reading an ROC curve. To do this, I'm going to present some "pre-canned" charts that will show extreme situations that should make it easier to understand what other ROC curves are "saying".

The first example is the simplest: a diagonal line. A diagonal line indicates that the classifier is just making completely random guesses. Since your classifier is only going to be correct 50% of the time, it stands to reason that your TPR and FPR will also be equal.

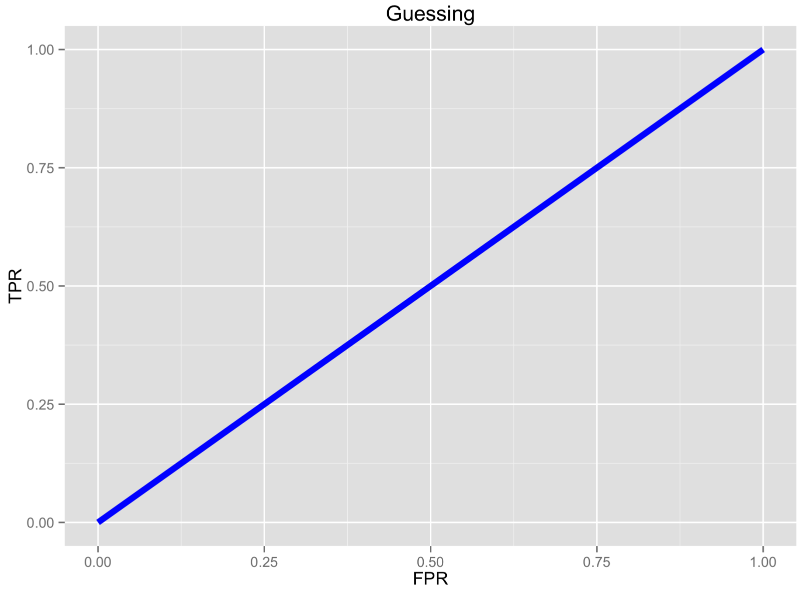


Figure 4: Guessing

Often times, ROC charts will include the random ROC curve to provide the user with a benchmark for what a naive classifier would do. Any curves above the line are better than guessing, while those below the line. . . well you're better off guessing.

We know what a totally random classifier looks like, but what about a PERFECT classifier – i.e. something that makes every prediction correctly. Well if you're lucky enough to have a perfect classifier, then you'll also have a perfect trade-off between TPR and FPR (meaning you'll have a TPR of 1 and an FPR of 0). In that case, your ROC curve looks something like this.

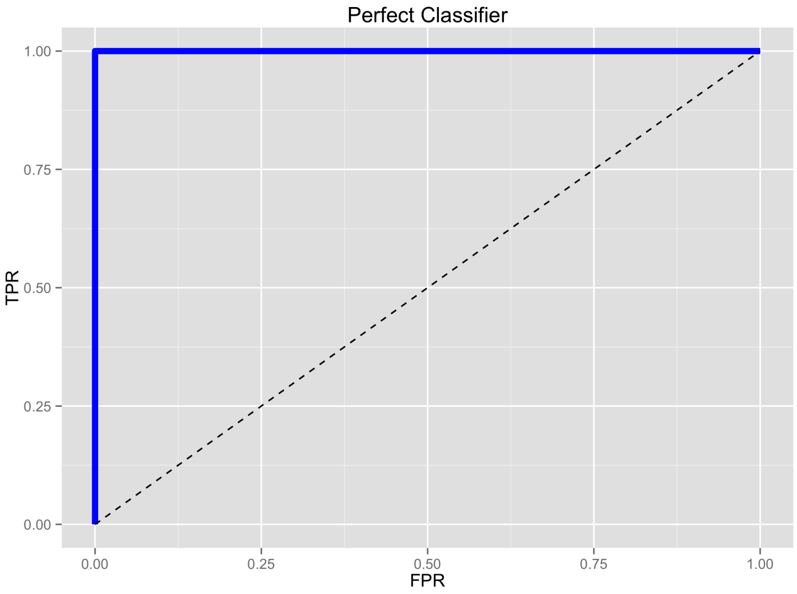


Figure 5: Perfect

So we know what a random classifier looks like and what a perfect classifier looks like, but what about a bad classifier? A bad classifier (i.e. something that's worse than guessing) will appear below the random line. This, my friend, is absolute garbage. Throw it away. . . now!

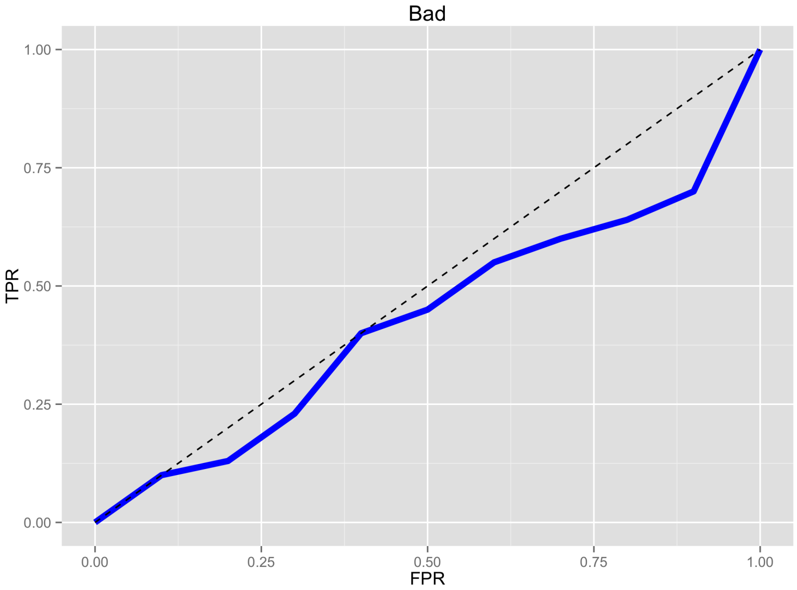


Figure 6: Bad

A much more interesting activity is attempting to decipher the difference between an "OK" and a "Good" classifier. The chart below shows an example of a very mediocre classifier. Context is everything of course, but there's not much lift here. In addition, be very wary of lines that dip or are very geometric looking. I've found that in practice, this can mean that there's an irregularity with your data, or you're making a very bad assumption in your model.

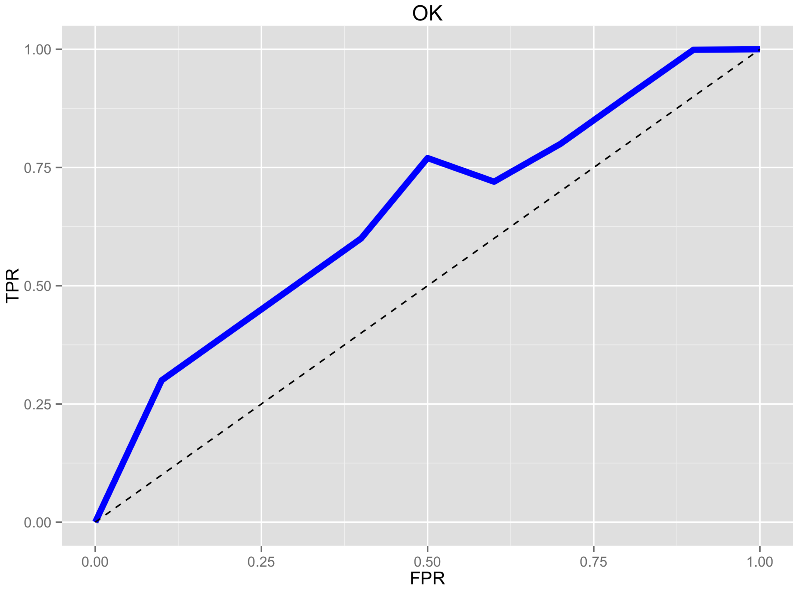


Figure 7: Ok

Ahh this is looking a little better. Below you can see a nice “hump shaped” (it’s a technical term) curve that’s continually increasing. It sort of looks like it’s being yanked up into that top left (the perfect) spot of the chart.

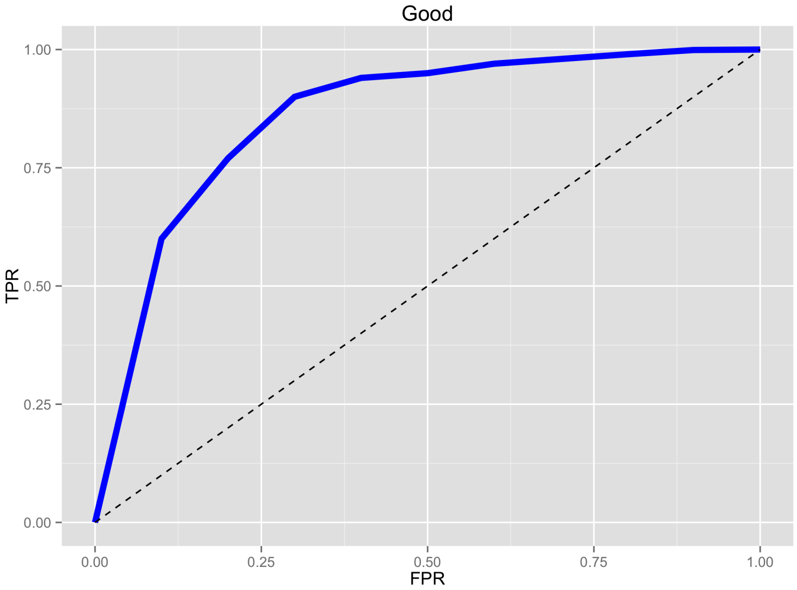


Figure 8: Good

So it turns out that the "hump shaped-ness" actually has a name: `AUC` or `Area Under the Curve`. One can't really give an overview of ROC curves without mentioning AUC. The good news is it's exactly what it sounds like – the amount of space underneath the ROC curve. You can think of the AUC as sort of a holistic number that represents how well your TPR and FPR is looking in aggregate.

To make it super simple:

- $AUC = 0$: VERY BAD
- $AUC = 1$: PERFECT

So in the context of an ROC curve, the more “up and left” it looks, the larger the AUC will be and thus, the better your classifier is. Comparing AUC values is also really useful when comparing different models, as we can select the model with the high AUC value, rather than just look at the curves.

- To Make ROC curve, one can use the `InformationValue::plotROC`. See R code.

Decide on Optimal Prediction Probability Cutoff for the Model

The paper mentioned that “Many of the incorrectly predicted unsuitable habitat cells fell just short of the 0.5 probability cut-off. If the cut-off point for the predicted suitable habitat were placed at 0.4, the model would have correctly identified 91 out of 121 grid cells (74% vs 58%) . . . ”. But, is there a way that can help us to determine the cut-off point? It is now helpful to plot the sensitivity vs the cutpoints overlayed with the specificity vs cutpoints.

4.2.3 Model Selection

The `frogs` data set has 212 rows and 11 columns. The data are on the distribution of the Southern Corroboree frog, which occurs in the Snowy Mountains area of New Wales, Australia. This data frame contains the following columns:

- `pres.abs` 0 = frogs were absent, 1 = frogs were present
- `northing` reference point
- `easting` reference point
- `altitude` altitude, in meters
- `distance` distance in meters to nearest extant population
- `NoOfPools` number of potential breeding pools
- `NoOfSites` number of potential breeding sites within a 2 km radius
- `avrain` mean rainfall for Spring period
- `meanmin` mean minimum Spring temperature
- `meanmax` mean maximum Spring temperature

Let's read in the data and take a look at it...

Effect Heredity

The principle of effect heredity relates to the **inclusion in the model of lower-order components of higher-order effects**. The motivation for this principle is observational evidence that

factors with small main effects tend not to have significant interaction effects.

- **Strong effect heredity** requires that all lower-order components of a model effect be included in the model. Suppose that a three-way interaction (ABC) is in the model. Then all of its component main effects and two-way interactions (A, B, C, AB, AC, BC) must also be in the model.
- **Weak effect heredity** requires that only a sequence of lower-order components of a model effect be included. If a three-way interaction is in the model, then the model must contain one of the factors involved and one two-way interaction involving that factor. Suppose that the three-way interaction ABC is in the model. Then if B and BC are also in the model, the model satisfies weak effect heredity.
- **For continuous factors, effect heredity ensures that the model is invariant to changes in the location and scale of the factors**.

7.3 Multinomial Logistic Regression

7.3.1 Multinomial Probability Distribution

The categorical response variable Y now has multiple levels $j = 1, \ldots, J$, where each category has $\pi_j = P(Y = j)$ and N_j is the number of trials responding with category j. These may or may not be ordinal responses. The PMF is known as the "multinomial probability distribution". (This is for a *single sample* of n observations.)

$$P(N_1 = n_1, \ldots, N_j = n_j) = \frac{n!}{\prod_{j=1}^{J} n_j!} \prod_{j=1}^{J} \pi_j^{n_j}$$

The MLE for each π_j is $\hat{\pi}_j = n_j/n$ (ie. proportion in each category).

7.3.2 Nominal Response Regression Models

We can define *odds* as a comparison of *any pair of response categories*; a popular regression model for multinomial resopnses is by forming the odds of the remaining $J - 1$ categories against a base category. This multinomial regression model looks like:

$$log(\pi_j/\pi_1) = \beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jk}x_k \; \forall j = 2, \ldots, J$$

Note that there is a separate set of β parameters *for each response* *category*, so each response's log-odds can relate to the explanatory variables in a different way. It's easy to compare other categories:

$$log\,(\pi_2/\pi_3) = log\,(\pi_2/\pi_1) - log\,(\pi_3/\pi_1) = log\,(\pi_2) - log\,(\pi_3) = (\beta_{20} - \beta_{30}) + (\beta_{21} - \beta_{3}$$

Calculating Probabilities

We can also calculate probabilities by maximum likelihood:

$$\pi_j = \pi_1 exp\,(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jk}x_k)\ \forall j = 2, \ldots, J$$

But we need to find π_1 first. Since $\pi_1 + \pi_2 + \cdots + \pi_J = 1$, we can get an expression for π_1:

$$\pi_1 = \frac{1}{1 + \sum_{j=2}^{J} exp\,(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jk}x_k)}$$

Combining, we get:

$$\pi_j = \frac{exp\,(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jk}x_k)}{1 + \sum_{j=2}^{J} exp\,(\beta_{j0} + \beta_{j1}x_1 + \cdots + \beta_{jk}x_k)} \quad \forall j = 2, \ldots, J$$

For a sample of size n observations, the likelihood function is the product of n multinomial distributions, with π_j as described in these equations. Iterative numerical procedures are used to find these MLEs using `nnet::multinom(...)`.

- See R code

We interpret these results as:

$$log\left(\hat{\pi}_{scab}/\hat{\pi}_{healthy}\right) = 30.55 - 0.65 \times I\left(class = SRW\right) - 21.60 \times density - 0.016 \times hard$$

$$log\left(\hat{\pi}_{sprout}/\hat{\pi}_{healthy}\right) = 19.17 - 0.22 \times I\left(class = SRW\right) - 15.12 \times density - 0.021 \times har$$

LRTs for significance of coefficients are straightforward using `Anova()`. Typically we want to explore whether a particular explanatory variable has an effect on *all* response categories (not just one). That hypothesis test is $H_0 : \beta_{jr} = 0 \ \ \forall j = 2, \ldots, J$.

Odds Ratios for Multinomial Models

Odds of a category j response vs. a category 1 response change by $exp\left(c\beta_{jr}\right)$ for every $c - unit$ change in x_r, holding other variables constant.

Odds of a category j response vs. a category j' response change by $exp[c(\beta_{jr} - \beta_{j'r})]$ for every $c - unit$ change in x_r, holding other variables constant.

Confidence Intervals for Parameters

The `confint()` function for multinomial regression does *not* use likelihood ratios; it uses Wald. We calculate CI for Odds Ratios similarly to binomial case, by finding the CI for the linear predictor first, and then exponentiating.

Note how confidence intervals are stored: [coefficients, lower:upper limits, class sequence from fit output].

7.4 Ordinal Logistic Regression

7.4.1 Proportional Odds Model for Ordinal Response Data

Basically, when we have ordinal data, we just apply the logit transformation to the *cumulative* probability distribution (CDF). The *cumulative* probability for Y is $P\,(Y|j) = \pi_1 + \pi_2 + \cdots + \pi_j$. Then we apply the logit transformation:

$$logit[P\,(Y \leq j)] = log\left[\frac{P\,(Y \leq j)}{1 - (Y \leq j)}\right] = log\left[\frac{\pi_1 + \cdots + \pi_j}{\pi_{j+1} + \cdots + \pi_J}\right]$$

Then the "**Proportional Odds Model**" applies a big simplification by assuming the β parameters *except for the intercept are the same for all explanatory variables* (for all $j = 1, \ldots, J-1$). This means that *effects of explanatory variables are the same, no matter which cumulative probabilities were used to form the log odds*! If we were to plot these logit curves, they would have the identical shape, just be shifted on the x-axis according to intercept terms.

$$logit[P\,(Y \leq j)] = \beta_{j0} + \beta_1 x_1 + \cdots + \beta_k x_k \;\; \forall J = 1, \ldots, J-1$$

Also, this can be rearranged to:

$$P\,(Y \leq j) = \frac{exp\,(\beta_{j0} + \beta_1 x_1 + \cdots + \beta_k x_k)}{1 + exp\,(\beta_{j0} + \beta_1 x_1 + \cdots + \beta_k x_k)} \;\; \forall J = 1, \ldots, J-1$$

It's important to remember that because of the nature of cumulative probabilities, $\beta_{J0} > \cdots > \beta_{10}$.

If we wanted a model where the β parameters were allowed to vary for level of J, that would be the "cumulative probability model". This model also gets more complicated when there are more than 2 categorical explanatory variables (because we need a m-dimensional contingency table).

We can also look at how we express π_j in terms of these cumulative probabilities:

$$\pi_j = P\,(Y = j) = P\,(Y \leq j) - P\,(Y \leq j-1) = \frac{exp\,(\beta_{j0} + \beta_1 x_1 + \cdots)}{1 + exp\,(\beta_{j0} + \beta_1 x_1 + \cdots)} - \frac{exp}{1 + e}$$

7.4.2 Estimation and Inference

Parameters are estimated using MLE, with the `MASS::polr(...)` function. If we have one explanatory variable, the hypotheses of interest are $H_0 : \beta_1 = 0$ and $H_a : \beta_1 \neq 0$.

If we reject H_0, then the ordering of log-odds comparing $P(Y \leq j)$ and $P(Y > j)$ holds. Log odds progressively grow larger (or smaller, depending on sign of β_1).

If we fail to reject H_0, then the log-odds comparing $PP(Y \leq j)$ and $P(Y > j)$ do not depend on this explanatory variable. For the case of two explanatory variables, this is equivalent to *independence*.

WARNING: levels of the Y variable must be properly ordered before using `polr(...)`. Also, all parameters generated by `polr(...)` need to have signs reversed, except for β_{j0}.

We can also predict probabilities and classes. ** However, `predict(...)` does not calculate the standard errors that would be needed in order to calculate Wald confidence intervals for π_j**.

7.4.3 Odds Ratio

Our odds ratio interpretation as $exp\,(\beta_{j0} + \beta_1 x_1 + \cdots + \beta_k x_k)$ is natural, except that we are using cumulative probabilities. Recall:

$$logit[P\,(Y \leq j)] = log\left[\frac{P\,(Y \leq j)}{1 - P\,(Y \leq j)}\right] = \beta_{j0} + \beta_1 x_1 + \cdots + \beta_k x_k$$

As usual, for a c-unit increase in x_1:

$$OR = \frac{Odds_{x_1+c}\,(Y \leq j)}{Odds_{x_1}\,(Y \leq j)} = exp\,(c\beta_1)$$

The interpretation is that the odds of $Y \leq j$ vs. $Y > j$ change by $exp\,(c\beta_1)$ for every c-unit increase in x_1, holding all other variables constant. Due to the nature of the proportional odds model, this is the same result *no matter which category is used for j*.

7.4.4 Case Study: Modeling an Ordinal Categorical Outcome in Ohio SMART

The Centers for Disease Control analyzes Behavioral Risk Factor Surveillance System (BRFSS) survey data for specific metropolitan and micropolitan statistical areas (MMSAs) in a program called the Selected Metropolitan/Micropolitan Area Risk Trends of BRFSS

(SMART BRFSS.)

In this work, we will focus on data from the 2016 SMART, and in particular on data from the Cleveland-Elyria, OH, Metropolitan Statistical Area. The purpose of this survey is to provide localized health information that can help public health practitioners identify local emerging health problems, plan and evaluate local responses, and efficiently allocate resources to specific needs.

The outcome we'll study now is `genhealth`, which has five ordered categories. I'll include the subset of all observations in `smart_ohio.csv` with complete data on these 7 variables.

Variable	Description
`SEQNO`	Subject identification code
`genhealth`	Five categories (1 = Excellent, 2 = Very Good, 3 = Good, 4 = Fair, 5 = Poor) on general health
`physhealth`	Now thinking about your physical health, which includes physical illness and injury, for how many days during the past 30 days was your physical health not good?

Variable	Description
`costprob`	1 indicates Yes to “Was there a time in the past 12 months when you needed to see a doctor but could not because of cost?”, and 0 otherwise.
`sleephrs`	average amount of sleep the subject gets in a 24-hour period
`incomegroup`	8 income groups from $<$ 10,000 to 75,000 or more
`bmi`	body-mass index

- See R code.