Lecture 3.1: Categorical Response Data

Applied Statistics and Data Analysis II

Jesús Arroyo

Setting

- Consider a sample of data $(y_1, X_1), \ldots, (y_n, X_n)$ such that:
  - Each X_i is a p-dimensional covariate vector.
  - The response y_i is a categorical variable in $\{0, 1, \ldots, K\}$.
- Set $\mathbf{y} = (y_1, \ldots, y_n)$ and $\mathbf{X} = [X_1, \ldots, X_n]^T \in \mathbb{R}^{n \times p}$.

- In particular, we consider three special response cases:
  - Binary response: $y_i \in \{0, 1\}$, which denotes a class label
  - Nominal responses: $y_i \in \{1, \ldots, K\}$ indicates a category or class, for which the magnitude of y_i is irrelevant.
  - Ordinal responses: $y_i \in \{1, \ldots, K\}$ such that $y_i > y_j$ indicates a level of preference, but the difference $y_i - y_j$ does not have a meaning.

Binary response data

Binary response data

- There are generally two different formats of binary response data:
  - Ungrouped data: each observation y_i corresponds to a single trial with outcome 0 or 1.
  - Grouped data: the response represents the proportion of positive trials in a set of observations that share the same explanatory variables.

Binary response data

- There are generally two different formats of binary response data:
  - Ungrouped data: each observation y_i corresponds to a single trial with outcome 0 or 1.
  - Grouped data: the response represents the proportion of positive trials in a set of observations that share the same explanatory variables.
- **Note:** grouped data is different from count data.

Example: ungrouped vs. grouped data

Value	Group Covariates	Individual Covariates	
$y_{1,1}$	X_1	$X'_{1,1}$	$\}\, n_1$
$y_{1,2}$	X_1	$X'_{1,2}$	
$\vdots$	$\vdots$	$\vdots$	
y_{1,n_1}	X_1	X'_{1,n_1}	
$y_{2,1}$	X_2	$X'_{2,1}$	$\}\, n_2$
$\vdots$	$\vdots$	$\vdots$	
$\vdots$	$\vdots$	$\vdots$	
$\vdots$	$\vdots$	$\vdots$	
$\vdots$	$\vdots$	$\vdots$	$\}\, n_m$
$\vdots$	$\vdots$	$\vdots$	
$y_{m,1}$	X_m	X'_{m,n_m}	

Ungrouped data

Group size	Proportion of successes	Group Covariates
n_1	$\frac{1}{n_1}\sum_{i=1}^{k_1} y_{1,i}$	X_1
n_2	$\frac{1}{n_2}\sum_{i=1}^{k_2} y_{2,i}$	X_2
$\vdots$	$\vdots$	$\vdots$
n_m	$\frac{1}{n_m}\sum_{i=1}^{k_m} y_{m,i}$	X_m

Grouped data

Bernoulli and binomial models

- Suppose that each $y_{i,k}$ is an independent Bernoulli random variable with $\mu_i = \mathbb{P}(y_{i,k} = 1)$, and define $\tilde{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{i,j}$. Then,

$$n_i \tilde{y}_i = y_{i,1} + \cdots + y_{i,n_i} \sim \text{Binomial}(n_i, \mu_i),$$

$$\mathbb{P}\left(\tilde{y}_i = \frac{k}{n_i}\right) = \binom{n_i}{k} \mu_i^k (1 - \mu_i)^{n_i - k}.$$

Bernoulli and binomial models

- Suppose that each $y_{i,k}$ is an independent Bernoulli random variable with $\mu_i = \mathbb{P}(y_{i,k} = 1)$, and define $\tilde{y}_i = \frac{1}{n_i} \sum_{j=1}^{n_i} y_{i,j}$. Then,

$$n_i \tilde{y}_i = y_{i,1} + \cdots + y_{i,n_i} \sim \text{Binomial}(n_i, \mu_i),$$

$$\mathbb{P}\left(\tilde{y}_i = \frac{k}{n_i}\right) = \binom{n_i}{k} \mu_i^k (1 - \mu_i)^{n_i - k}.$$

- Using this equivalence, it is always possible to express a binomial GLM for grouped data with a binary GLM for the corresponding ungrouped data. The estimated coefficients $\hat{\boldsymbol{\beta}}$ are the same.
- On the other hand, in the presence of individual covariates, it is not possible to aggregate ungrouped data into groups.

Binomial GLM model

- The log-likelihood of a binomial GLM can be expressed as

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{m} \left[\tilde{y}_i n_i \log\left(\frac{\mu_i}{1-\mu_i}\right) + n_i \log(1-\mu_i) + \log\binom{n_i}{\tilde{y}_i} \right],$$

  with $g(\mu_i) = X_i^T \boldsymbol{\beta}$.
- If $n_i = 1$ for all $i = 1, \ldots, m$, then the log-likelihood is just a binary GLM.
- Results for binary GLMs also apply to binomial GLMs (with small modifications in the formulas to include $n_1, \ldots, n_m$).
- One main difference: if each $n_i \to \infty$, $i = 1, \ldots, m$ and $n_i \mu_i (1-\mu_i) \to \infty$, then the deviance (equiv. scaled deviance) converges to a χ^2_{m-p} distribution.

GLMs for binary data

- Consider binary ungrouped data $(y_1, X_1), \ldots, (y_n, X_n)$.
- The log-likelihood of a binary GLM can be expressed as

$$\ell(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i \log\left(\frac{\mu_i}{1-\mu_i}\right) + \log(1-\mu_i) \right].$$

- Three popular choices for the link function $g(\mu_i) = X_i^T \boldsymbol{\beta}$:
  - Probit: when g^{-1} is the cumulative distribution function of a standard normal distribution Φ. If $Z \sim N(0,1)$, $\Phi(x) = \mathbb{P}(Z \leq x)$. Hence,
  $$\mathbb{P}(y_i = 1 | X_i) = \Phi(X_i^T \boldsymbol{\beta}).$$
  - Logit: this is the log-odds link function that we studied before.
  - Linear: g is the identity link function, i.e., $\mu_i = X_i^T \boldsymbol{\beta}$, and
  $$\mathbb{P}(y_i = 1 | X_i) = X_i^T \boldsymbol{\beta}.$$

Logistic regression

Logistic regression

- Recall that a **logistic regression** model is a binary GLM that uses the logit (or the log-odds) link function, which is the canonical link function of the binary GLM, and it is given by

$$\text{logit}(\mu_i) = \log\left(\frac{\mu_i}{1-\mu_i}\right).$$

- Hence, the probability that y_i is equal to 1 can be written as

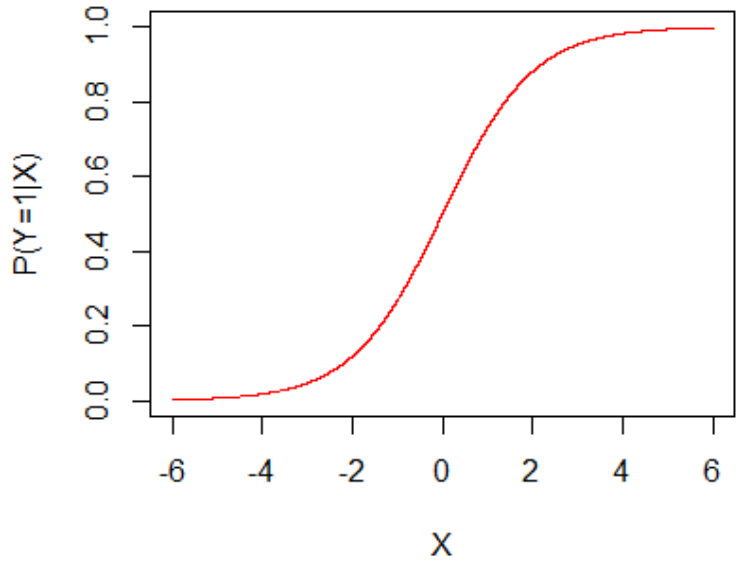
$$\mu_i = \mathbb{P}(y_i = 1|X_i) = \frac{\exp\left(X_i^T\boldsymbol{\beta}\right)}{1+\exp\left(X_i^T\boldsymbol{\beta}\right)}.$$

Interpretation of the coefficients

- When there is a single quantitative variable in the model $x_i \in \mathbb{R}$, the curve of μ_i takes the form of a logisic function

$$\mu_i = \frac{1}{1 + \exp(-\beta x_i)}.$$

- If there are multiple variables, the marginal plot of each variable (or any linear combination) has the same shape.



Interpretation of the coefficients

How do we interpret β_j when there are p variables in the model?

- If $\beta_j = 0$, then variable j is **conditionally independent** of the response given the rest of the variables.

Interpretation of the coefficients

How do we interpret β_j when there are p variables in the model?

- If $\beta_j = 0$, then variable j is **conditionally independent** of the response given the rest of the variables.
- More generally, consider X_i and $\tilde{X}_i$ such that for $k = 1, \ldots, p$,

$$\tilde{X}_{ik} = \begin{cases} X_{ik} & \text{if } k \neq j, \\ X_{ik} + 1 & \text{if } k = j. \end{cases}$$

  That is, $\tilde{X}_i$ increases one unit the value of variable j while keeping the rest of the variables fixed.

- Observe that

$$\begin{aligned} \beta_j &= \tilde{X}_i^T \boldsymbol{\beta} - X_i^T \boldsymbol{\beta} \\ &= \text{logit}\big(\mathbb{P}(y_i = 1|\tilde{X}_i)\big) - \text{logit}\big(\mathbb{P}(y_i = 1|X_i)\big). \end{aligned}$$

Interpretation of the coefficients

- By exponentiating both sides of the previous expression, we obtain

$$e^{\beta_j} = \frac{\frac{\mathbb{P}(y_i=1|\tilde{X}_i)}{1-\mathbb{P}(y_i=1|\tilde{X}_i)}}{\frac{\mathbb{P}(y_i=1|X_i)}{1-\mathbb{P}(y_i=1|X_i)}}.$$

  The right-hand side is an odds ratio.
- Hence, increasing by one unit the value of the variable j (while keeping the rest of the variables constant), the odds change by a factor of e^{β}.

Interpretation of the coefficients

- By exponentiating both sides of the previous expression, we obtain

$$e^{\beta_j} = \frac{\frac{\mathbb{P}(y_i=1|\tilde{X}_i)}{1-\mathbb{P}(y_i=1|\tilde{X}_i)}}{\frac{\mathbb{P}(y_i=1|X_i)}{1-\mathbb{P}(y_i=1|X_i)}}.$$

  The right-hand side is an odds ratio.
- Hence, increasing by one unit the value of the variable j (while keeping the rest of the variables constant), the odds change by a factor of e^{β}.
- **Example**: an odds value of of 5 indicates that the probability of $y = 1$ is five times larger than the probability of $y = 0$. If $\beta_j = \ln 2$, then a one-unit change in variable j (keeping the other variables constant) will result in an odds value of 10.

Example: spam classification

- The spam dataset contains information about a collection of emails, some of which are classified as spam. The full data can be downloaded from:

  `https://archive.ics.uci.edu/ml/datasets/spambase`

- Variables in the data represent the percentage of words or characters in an email that match a certain value. Here we only consider eight of them:
  - `mail.`
  - `free.`
  - `credit.`
  - `$.`
  - `edu.`
  - `project.`
  - `conference.`
  - `semicolon.`

Example: spam classification

```
> spam.glm <- glm(formula = spam ~., family = "binomial", data = spam.data)
Warning message:
glm.fit: fitted probabilities numerically 0 or 1 occurred
> summary(spam.glm)


Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept)        -1.21226    0.04961 -24.435  < 2e-16 ***
word_freq_mail      0.25892    0.06232   4.155 3.25e-05 ***
word_freq_free      1.40962    0.10139  13.903  < 2e-16 ***
word_freq_credit    2.89501    0.39074   7.409 1.27e-13 ***
char_freq_dollar   11.91458    0.61430  19.395  < 2e-16 ***
word_freq_edu      -1.93478    0.26857  -7.204 5.85e-13 ***
word_freq_project  -2.71333    0.44843  -6.051 1.44e-09 ***
word_freq_conference -3.96353    1.03129  -3.843 0.000121 ***
char_freq_semicolon  -1.24374    0.48223  -2.579 0.009904 **
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Example: spam data

Consider increasing in one unit the percentage of the word “free” in the first sample of the data.

```
x1 <- spam.data[1,]
x2 <- x1
x2[2] <- x1[2] + 1
```

	x1	x2
word_freq_mail	0.00	0.00
word_freq_free	0.32	1.32
word_freq_credit	0.00	0.00
char_freq_dollar	0.00	0.00
word_freq_edu	0.00	0.00
word_freq_project	0.00	0.00
word_freq_conference	0.00	0.00
char_freq_semicolon	0.00	0.00

Example: spam data

```
# Predicted probability of Y given x1
> p1 <- predict(spam.glm, newdata = x1, type = "response")
0.3183892

> odds1 <- p1/(1-p1)
0.4671129

# Increment on the odds
> exp.beta.j <- exp(spam.glm$coefficients[3])
4.094401

# Predicted probability of Y given x2
> p2 <- (exp.beta.j*p1/(1-p1)) / (1 + exp.beta.j * p1/(1-p1))
0.656658

> predict(spam.glm, newdata = x2, type = "response")
0.656658


> odds2 <- p2 / (1-p2)
1.912548
```

Inference about parameters

- In the previous lectures, we have established the basis for statistical inference in GLMs, and the same results apply for logistic regression.
- The MLE $\hat{\boldsymbol{\beta}}$ has a large-sample normal distribution around $\boldsymbol{\beta}$. The covariance matrix is equal to the inverse of the information matrix $\mathbf{J} = \mathbf{X}^T\mathbf{W}\mathbf{X}$, where $\mathbf{W}$ is diagonal, and

$$\mathbf{W}_{ii} = \mu_i(1 - \mu_i).$$

- For hypothesis testing, we can use the Wald, likelihood ratio or score test statistics. However, when the effect size is large, the Wald method is less powerful (i.e., the probability of rejecting the null when the null is false is smaller than the other tests), or can even fail.

Predictive power

- Logistic regression is popularly used in statistics and machine learning for binary classification.
- To construct a classifier, logistic regression introduces a cutoff $\hat{\pi}$ to define the estimated class as $\hat{y}_i = 1$ if $\hat{\mu}_i > \hat{\pi}$, and $\hat{y}_i = 0$ otherwise.
- The predictive power of a classifier can be measured in different ways, such as:
  - Classification error (or accuracy).
  - Receiving operating characteristic (ROC) curve.

Classification error

- The **classification error** is an estimate of the expected risk under a 0-1 loss function. More generally, the loss function can penalize differently the error for each class.
- Logistic regression estimates two different parameters ($\hat{\boldsymbol{\beta}}$ and $\hat{\pi}$) for which two different sets of data are required to fit the model:
  - A training set is used to fit the coefficients $\hat{\boldsymbol{\beta}}$.
  - A validation set is used to choose the classification cutoff $\hat{\pi}$.
- To obtain an unbiased estimator of the classification error, we can use a test set of data to evaluate the error.
- Alternatively, a *nested cross-validation* strategy can be employed if the sample size is small.

Classification error

- For a given X, logistic regression can estimate the probability that its corresponding label y is equal to 1 as,

$$\hat{\mu} = (1 + \exp(-X\hat{\boldsymbol{\beta}}))^{-1}.$$

- A classification rule $C(X)$ for X is defined using the cutoff $\hat{\pi}$ by

$$C(X) = \hat{y} = \mathbb{1}(\hat{\mu} > \hat{\pi}).$$

- The cutoff is selected to minimize the error on the validation set.
- Two common choices that do not require a validation set are $\hat{\pi} = 0.5$ or $\hat{\pi} = \frac{1}{n}\sum_{i=1}^{n} y_i$ (the average of $\mathbf{y}$ on the training set).

Example: spam classification

```
spam.data <- read.csv("spam-data.csv")

# Split training - validation - test
# 60% training 20% validation 20% test
n <- nrow(spam.data)   # n = 4601

training.validation <- sample(1:n, size = 0.8*n)
test <- (1:n)[-training.validation]
training <- sample(training.validation, size = 0.6*n)
validation <- (1:n)[-c(test, training)]

# glm fit using training
spam.glm <- glm(formula = spam ~., family = "binomial",
                data = spam.data, subset = training)
```

Example: spam classification

```
> # Estimated probabilities on  thevalidation set
> mu_hat_validation <- predict(spamsub.glm, newdata = spam.data.sub[validation,],
+                              type = "response")

> y_training <-spam.data[training,]$spam
> y_validation <- spam.datasub[validation,]$spam
> y_test <- spam.data.sub[test,]$spam

> # cutoff
> pi_0 <- mean(y_training)
0.3894928

> y_hat_validation <- 1*(mu_hat_validation > pi_0)

> table(y_validation, y_hat_validation)
            y_hat_validation
y_validation   0   1
           0 500  33
           1 112 275

> #classification error on the validation set
> sum(y_validation != y_hat_validation) / length(y_validation)
0.1576087
```

Example: spam data

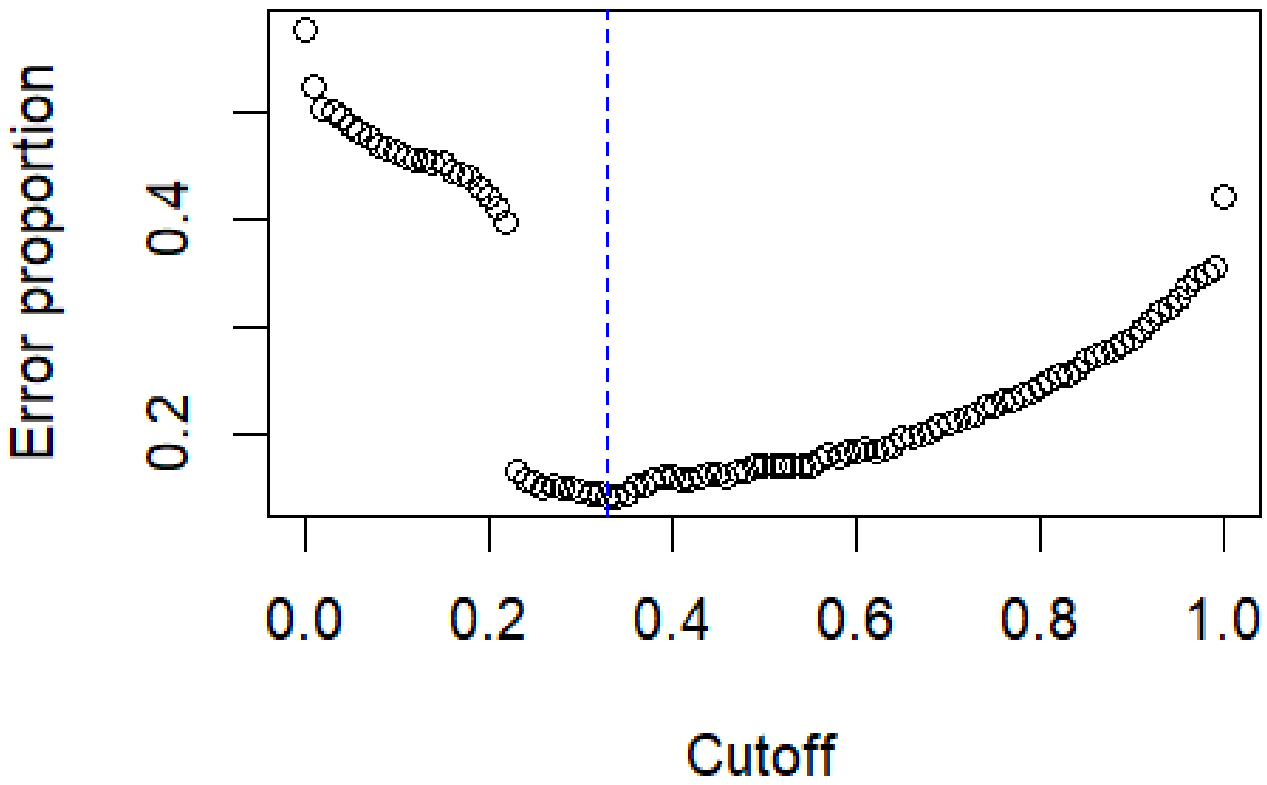
```
# Function to calculate classification error per cutoff
class_error <- function(cutoff) {
y_hat_validation <- 1*(mu_hat_validation > cutoff)
error <- sum(y_validation != y_hat_validation) / length(y_validation)
return(error)
}

# Grid of cutoffs
cutoffs.seq <- seq(0, 1, 0.01)

# Calculate error for each cutoff
error.per.cutoff <- sapply(cutoffs.seq, class_error)
best.cutoff <- cutoffs.seq[which.min(error.per.cutoff)]
```

Example: spam data

```
> plot(cutoffs.seq, error.per.cutoff, xlab = "Cutoff", ylab = "Error proportion")
> abline(v = best.cutoff, lty=2, col="blue")
> best.cutoff
0.33
```



Example: spam data

The training set was used to estimate $\hat{\beta}$, and the validation set to estimate the cutoff $\hat{\pi}$. We use the test set to obtain an unbiased estimate of the classification error.

```
> # Fitted probabilities on the test set
> mu_hat_test <- predict(spam.glm, newdata = spam.data.sub[test,],
+                        type = "response")
> # Estimated class labels
> y_hat_test <- 1*(mu_hat_test > best.cutoff)

> # Classification error
> sum(y_test != y_hat_test) / length(y_test)
[1] 0.1563518

> table(y_test, y_hat_test)
         y_hat_test
    y_test   0   1
         0 503  67
         1  77 274
```

ROC curve

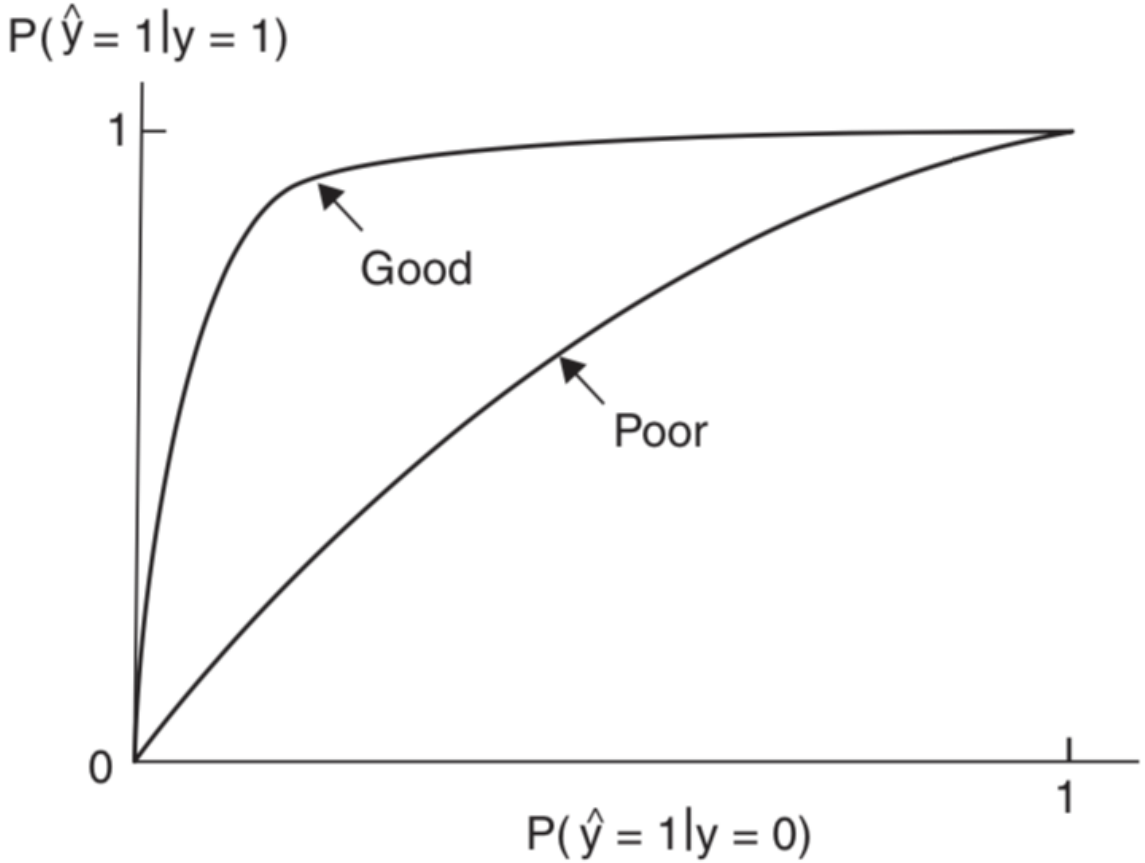
- The classification error is not very informative when the classes are imbalanced.
- Instead, we can consider the accuracy of the classifier for all the different cutoffs.
- The sensitivity and specificity are defined as

$$\text{sensitivity} = \mathbb{P}(\hat{y} = 1 | y = 1),$$
$$\text{specificity} = \mathbb{P}(\hat{y} = 0 | y = 0).$$

- These values can be estimated for any cutoff using the validation set.
- The sensititivity is *the true positive rate* (TPR), and $(1 - \text{specificity})$ is the *false positive rate* (FPR).

ROC curve

- The **receiver operating characteristic (ROC) curve** plots the FPR and TPR for different cutoffs.
- The area under the ROC curve gives a measure of predictive performance.



Example: spam data

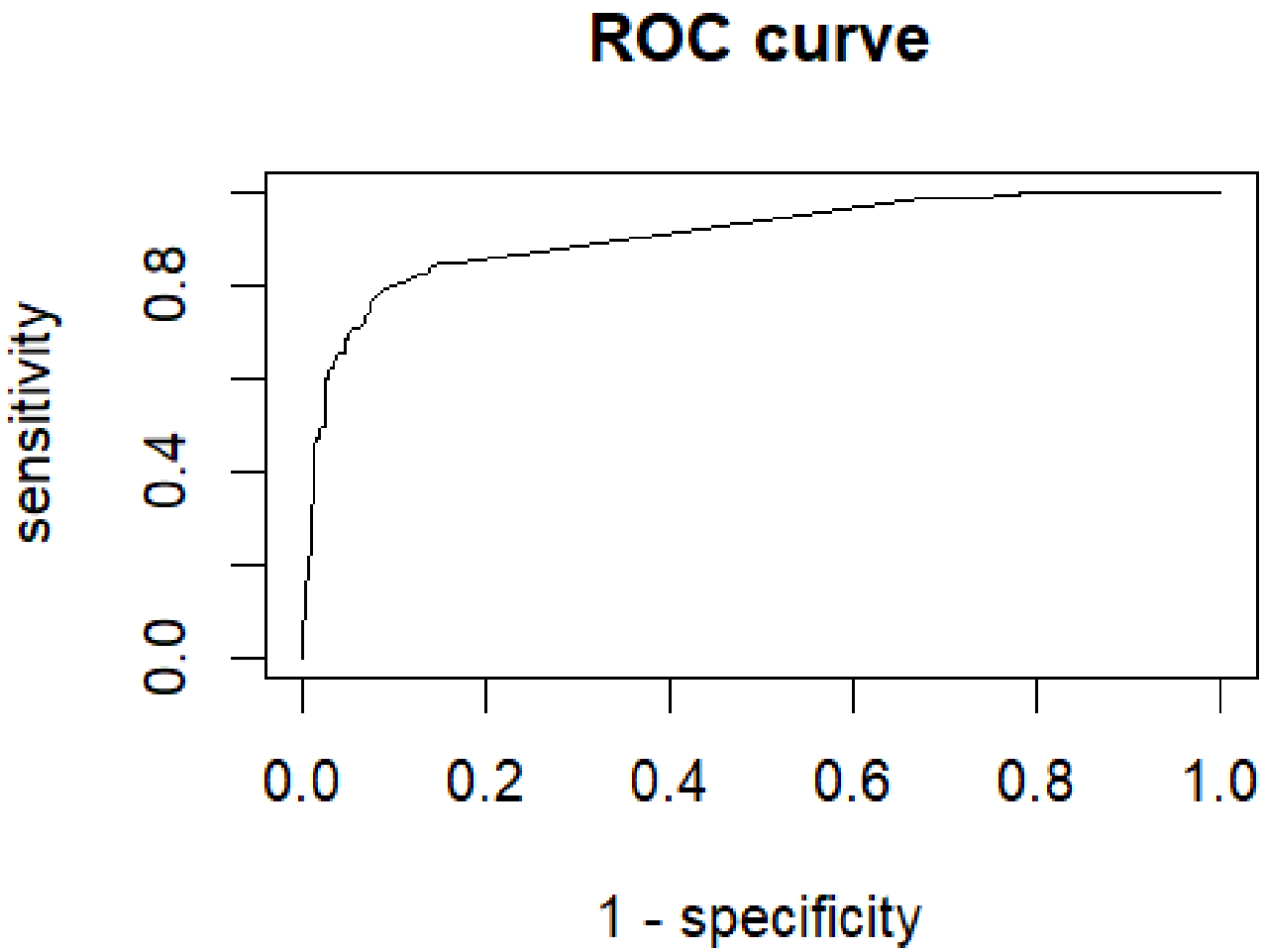
```
# specificity vs sensitivity
calculate.sensitivity <- function(cutoff) {
    which.Y <- which(y_validation == 1)
    y_hat_validation <- 1*(mu_hat_validation > cutoff)
    sensit <- sum(y_hat_validation[which.Y] == 1) / length(which.Y)
    return(sensit)
}

calculate.specificity <- function(cutoff) {
    which.Y <- which(y_validation == 0)
    y_hat_validation <- 1*(mu_hat_validation > cutoff)
    spec <- sum(y_hat_validation[which.Y] == 0) / length(which.Y)
    return(spec)
}


sensitivity <- sapply(cutoffs.seq, calculate.sensitivity)
specificity <- sapply(cutoffs.seq, calculate.specificity)
```

Example: spam data

```
> plot(1-specificity, sensitivity, type = "l", main = "ROC curve")
```



Choosing a cutoff for classification

In practice, selecting the cutoff to define a classifier depends on the final goal:

- **Minimize the classification error.** If the goal is to make the smallest number of errors on average (or to minimize a function of the errors), then the cutoff can be chosen by minimizing the error on the validation set (or by doing cross-validation).
- **Control the percentage of false positives.** When the goal is to limit the percentage of errors for a certain class by some $\epsilon > 0$ (for example, to limit the amount of important emails that are classified as spam), then the cutoff can be selected as the point in the ROC curve with the largest TPR and with FPR$\leq \epsilon$.

Multinomial response

Multinomial response variables

- In this part, we consider a categorical response variable with K different categories or classes.
- For the sample i, the response is encoded using a K-dimensional vector

$$\mathbf{y}_i = (y_{i1}, \ldots, y_{iK}),$$

  where y_{ij} indicates a value for category j in sample i:
  - For ungrouped data, $y_{ij} = 1$ if sample i belongs to class j, and $y_{ij} = 0$ otherwise. Therefore, $\sum_{j=1}^{K} y_{ij} = 1$.
  - For grouped data, each y_{ij} counts the number of observations in class j, and $n_i = \sum_{j=1}^{K} y_{ij}$.
- In this section, we consider data $(\mathbf{y}_1, X_1), \ldots, (\mathbf{y}_n, X_n)$, with $X_i \in \mathbb{R}^p$.

Example: 2016 presidential election results

- The file `2016-election-county.csv` (on Blackboard) contains the number of votes by county for each candidate (Clinton, Trump or Other) during the 2016 US presidential election[1].
- The data also contain demographic statistics by county.
- We model the number of votes for each candidate using the county information. Note that the county facts are not necessarily the same as the characteristics of the voting population.
- The response is a multi-categorical $(K = 3)$ variable counting the number of votes for each option.

[1] Data obtained from `https://www.kaggle.com/joelwilson/2012-2016-presidential-elections`.

Example: 2016 presidential election results

- We only consider the following covariates:
  - `State`: categorical.
  - `Age`: population by range of age (%)
  - `Female` (%)
  - `Race`: population that identify with a certain race (%)
  - `Foreignborn`: population born in a foreign country (%)
  - `Education`: cumulative level of education completed (%)
  - `Homeownership`: population that owns a house (%)
  - `Income`: per-capita income per year.
  - `Poverty`: persons below poverty level (%)
  - `Density`: population per square mile.

Example: 2016 presidential election results

- We only consider the following covariates:
  - `State`: categorical.
  - `Age`: population by range of age (%)
  - `Female` (%)
  - `Race`: population that identify with a certain race (%)
  - `Foreignborn`: population born in a foreign country (%)
  - `Education`: cumulative level of education completed (%)
  - `Homeownership`: population that owns a house (%)
  - `Income`: per-capita income per year.
  - `Poverty`: persons below poverty level (%)
  - `Density`: population per square mile.

- **Note:** in general, when the covariates are categorical (either ungrouped, e.g. state, or grouped, e.g. sex, race, education, etc.), one of the levels is used as a reference, and the results should be interpreted with respect to the reference level.

Example: 2016 presidential election results

```
> election.data <- read.csv("../../data/2016-election-county.csv", header = TRUE)
> summary(election.data)
```

```
   Clinton             Trump              Other           state_abbr     Age_under5        Age_5_17
Min.   :      4   Min.   :    57   Min.   :     3   TX      : 254   Min.   : 1.500   Min.   : 7.40
1st Qu.:   1166   1st Qu.:  3206   1st Qu.:   165   GA      : 159   1st Qu.: 5.200   1st Qu.:20.50
Median :   3153   Median :  7164   Median :   442   VA      : 133   Median : 5.800   Median :22.40
Mean   :  20019   Mean   : 19600   Mean   :  2058   KY      : 120   Mean   : 5.887   Mean   :22.54
3rd Qu.:   9600   3rd Qu.: 17427   3rd Qu.:  1422   MO      : 115   3rd Qu.: 6.500   3rd Qu.:24.20
Max.   :1893770   Max.   :620285   Max.   :138017   KS      : 105   Max.   :13.300   Max.   :40.50
                                                    (Other):2226
```

```
  Age_65plus        Female          White           Black       Foreignborn     Edu_highschool
Min.   : 4.10   Min.   :30.10   Min.   : 5.9   Min.   : 0.0   Min.   : 0.000   Min.   :45.00
1st Qu.:14.80   1st Qu.:49.50   1st Qu.:80.7   1st Qu.: 0.8   1st Qu.: 1.200   1st Qu.:80.10
Median :17.30   Median :50.40   Median :92.0   Median : 2.4   Median : 2.500   Median :85.90
Mean   :17.64   Mean   :49.96   Mean   :85.4   Mean   : 9.3   Mean   : 4.493   Mean   :84.51
3rd Qu.:19.93   3rd Qu.:51.10   3rd Qu.:96.0   3rd Qu.:10.9   3rd Qu.: 5.525   3rd Qu.:89.70
Max.   :52.90   Max.   :56.80   Max.   :99.3   Max.   :85.1   Max.   :51.300   Max.   :99.00
```

```
 Edu_bachelor   Home_ownership      Income         Poverty          Density
Min.   : 3.20   Min.   :19.40   Min.   : 8768   Min.   : 0.90   Min.   :    0.10
1st Qu.:13.70   1st Qu.:68.40   1st Qu.:19897   1st Qu.:12.18   1st Qu.:   17.68
Median :17.50   Median :73.45   Median :22889   Median :16.00   Median :   45.70
Mean   :19.74   Mean   :72.27   Mean   :23565   Mean   :16.71   Mean   :  261.54
3rd Qu.:23.40   3rd Qu.:77.60   3rd Qu.:26187   3rd Qu.:20.30   3rd Qu.:  115.10
Max.   :74.40   Max.   :93.80   Max.   :62498   Max.   :53.20   Max.   :69467.50
```

Example: 2016 presidential election results

```
> head(election.data)
```

	Clinton	Trump	Other	state_abbr	Age_under5	Age_5_17	Age_65plus	Female	White	Black
1	5908	18110	643	AL	6.0	25.2	13.8	51.4	77.9	18.7
2	18409	72780	2901	AL	5.6	22.2	18.7	51.2	87.1	9.6
3	4848	5431	111	AL	5.7	21.2	16.5	46.6	50.2	47.6
4	1874	6733	141	AL	5.3	21.0	14.8	45.9	76.3	22.1
5	2150	22808	426	AL	6.1	23.6	17.0	50.5	96.0	1.8
6	3530	1139	32	AL	6.3	21.4	14.9	45.3	26.9	70.1

	Foreignborn	Edu_highschool	Edu_bachelor	Home_ownership	Income	Poverty	Density
1	1.6	85.6	20.9	76.8	24571	12.1	91.8
2	3.6	89.1	27.7	72.6	26766	13.9	114.6
3	2.9	73.7	13.4	67.7	16829	26.7	31.0
4	1.2	77.5	12.1	79.0	17427	18.1	36.8
5	4.3	77.0	12.1	81.0	20730	15.8	88.9
6	5.4	67.8	12.5	74.3	18628	21.6	17.5

Multinomial distribution

- The **multinomial distribution** is a generalization of the binomial distribution that models a random sampling of n_i elements from K different classes that are chosen independently with replacement.

- Denote by $\mu_{i1}, \ldots, \mu_{jK}$ to the class probabilities for the observation i, with $\sum_{j=1}^{K} \mu_{ij} = 1$. The vector $\mathbf{y}_i$ has a multinomial distribution

$$\mathbf{y}_i \sim \text{Multinomial}(n_i, \mu_{i1}, \ldots, \mu_{ik})$$

if the probability mass function for $(h_1, \ldots, h_K)$, with $\sum_{j=1}^{K} h_j = n_i$, is given by

$$\mathbb{P}(\mathbf{y}_i = (h_1, \ldots, h_K)) = \frac{n_i!}{h_1! \cdots h_K!} \mu_1^{h_1} \cdots \mu_K^{h_K}.$$

Multinomial distribution: properties

- If $K = 2$, then $\mathbf{y}_i = (y_{i1}, n_i - y_{i1})$, and y_{i1} has a binomial distribution

$$y_{i1} \sim \text{Binomial}(n_i, \mu_{i1}).$$

- More generally, the marginal distribution of a sum of entries $\sum_{j \in \mathcal{J}} y_{ij}$, with $\mathcal{J} \subset \{1, \ldots, K\}$, is binomial distributed with parameters n_i and $\sum_{j \in \mathcal{J}} \mu_{ij}$. Therefore, aggregating the values of $\mathbf{y}$ on different classes results in an equivalent binomial model.

Multinomial distribution: properties

- If $K = 2$, then $\mathbf{y}_i = (y_{i1}, n_i - y_{i1})$, and y_{i1} has a binomial distribution

$$y_{i1} \sim \text{Binomial}(n_i, \mu_{i1}).$$

- More generally, the marginal distribution of a sum of entries $\sum_{j \in \mathcal{J}} y_{ij}$, with $\mathcal{J} \subset \{1, \ldots, K\}$, is binomial distributed with parameters n_i and $\sum_{j \in \mathcal{J}} \mu_{ij}$. Therefore, aggregating the values of $\mathbf{y}$ on different classes results in an equivalent binomial model.

- Similarly, if $\mathbf{u}_1, \ldots, \mathbf{u}_m$ are independent random vectors with distribution Multinomial$(1, \mu_1, \ldots, \mu_n)$, then

$$\mathbf{v} = \mathbf{u}_1 + \cdots + \mathbf{u}_m \sim \text{Multinomial}(m, \mu_1, \ldots, \mu_n),$$

  showing the equivalence between modelling grouped and independent ungrouped observations.

Baseline-category logits

- Consider ungrouped data on K categories (multiclass classification): $(\mathbf{y}_1, X_1), \ldots, (\mathbf{y}_n, X_n)$, with $\sum_{j=1}^{K} y_{ik} = 1$.
- We model the data using a multinomial distribution with parameters $\boldsymbol{\mu}_i = (\mu_{i1}, \ldots, \mu_{iK})$ for each observation $i = 1, \ldots, n$.
- Because the class probabilities are fully specified using only $K - 1$ classes, one of the classes is chosen as a **baseline**. By default, we will assume that this is class K, and write

$$\mu_{iK} = 1 - (\mu_{i1} + \mu_{i2} + \cdots + \mu_{i,K-1}) .$$

Baseline-category logits

- Consider ungrouped data on K categories (multiclass classification): $(\mathbf{y}_1, X_1), \ldots, (\mathbf{y}_n, X_n)$, with $\sum_{j=1}^{K} y_{ik} = 1$.
- We model the data using a multinomial distribution with parameters $\boldsymbol{\mu}_i = (\mu_{i1}, \ldots, \mu_{iK})$ for each observation $i = 1, \ldots, n$.
- Because the class probabilities are fully specified using only $K - 1$ classes, one of the classes is chosen as a **baseline**. By default, we will assume that this is class K, and write

$$\mu_{iK} = 1 - (\mu_{i1} + \mu_{i2} + \cdots + \mu_{i,K-1}) .$$

- Each response category is paired with the baseline category using

$$\log\left(\frac{\mu_{i1}}{\mu_{iK}}\right), \log\left(\frac{\mu_{i2}}{\mu_{iK}}\right), \ldots, \log\left(\frac{\mu_{i,K-1}}{\mu_{iK}}\right).$$

  These are called **baseline-category logits**.

Baseline-category logits

- When $K = 2$, the baseline-category logit is just the log-odds of the probability of being on class one.
- For $K > 2$, the baseline-category logit can be interpreted as the log-odds of a conditional probability:

$$\begin{aligned}\log\left(\frac{\mu_{ij}}{\mu_{iK}}\right) &= \log\left[\frac{\mathbb{P}(y_{ij}=1|y_{ij}=1 \text{ or } y_{iK}=1, X_i)}{1-\mathbb{P}(y_{ij}=1|y_{ij}=1 \text{ or } y_{iK}=1, X_i)}\right] \\ &= \text{logit}\left[\mathbb{P}(y_{ij}=1|y_{ij}=1 \text{ or } y_{iK}=1, X_i)\right].\end{aligned}$$

This is the log-odds that sample i is in class j, conditioned on the event that the sample is either in class j or K.

Multinomial logistic regression

- The **multinomial logistic regression model** simultaneously describes the effect of X_i on the $K-1$ baseline-category logits using $K-1$ different vectors $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_{K-1}$ such that $\boldsymbol{\beta}_j = (\beta_{ji}, \ldots, \beta_{jp})^T$, and

$$\log\left(\frac{\mu_{ij}}{\mu_{iK}}\right) = X_i^T \boldsymbol{\beta}_j, \qquad j = 1, \ldots, K-1,$$

$$\mu_{i1} + \cdots + \mu_{i,K-1} + \mu_{iK} = 1.$$

Multinomial logistic regression

- The value of the coefficients depend on the category chosen as a baseline, but these can be related easily for any classes a and b using the identity

$$\log\left(\frac{\mu_{ia}}{\mu_{ib}}\right) = \log\left(\frac{\mu_{ia}}{\mu_{iK}}\right) - \log\left(\frac{\mu_{ib}}{\mu_{iK}}\right) = X_i^T(\boldsymbol{\beta}_a - \boldsymbol{\beta}_b).$$

  Thus, the models are equivalent regardless of the baseline choice.
- For model identifiability, we define $\boldsymbol{\beta}_K = \mathbf{0}$.

Multinomial logistic regression

- The value of the coefficients depend on the category chosen as a baseline, but these can be related easily for any classes a and b using the identity

$$\log\left(\frac{\mu_{ia}}{\mu_{ib}}\right) = \log\left(\frac{\mu_{ia}}{\mu_{iK}}\right) - \log\left(\frac{\mu_{ib}}{\mu_{iK}}\right) = X_i^T(\boldsymbol{\beta}_a - \boldsymbol{\beta}_b).$$

  Thus, the models are equivalent regardless of the baseline choice.
- For model identifiability, we define $\boldsymbol{\beta}_K = \mathbf{0}$.

- The response probabilities can be calculated for any class $j = 1, \ldots, K$ as

$$\mu_{ij} = \frac{\exp\left(X_i^T\boldsymbol{\beta}_j\right)}{1 + \sum_{h=1}^{K-1}\exp(X_i^T\boldsymbol{\beta}_h)} = \frac{\exp\left(X_i^T\boldsymbol{\beta}_j\right)}{\sum_{h=1}^{K}\exp(X_i^T\boldsymbol{\beta}_h)}.$$

Model interpretation

- The interpretation of the coefficients is similar to the binary logistic regression, but in terms of the baseline category.
- Consider $\tilde{X}_i$ and X_i such that $\tilde{X}_{ih} = X_{ih} + 1$ in variable h (and equal otherwise). The baseline-category logits are related by

$$\exp(\beta_{jh}) \left(\frac{\mathbb{P}(y_{ij} = 1|X_i)}{\mathbb{P}(y_{iK} = 1|X_i)} \right) = \left(\frac{\mathbb{P}(y_{ij} = 1|\tilde{X}_i)}{\mathbb{P}(y_{iK} = 1|\tilde{X}_i)} \right).$$

- Therefore, $\exp(\beta_{jh})$ is a factor that indicates how much the odds of μ_{ij} and μ_{iK} change by increasing in one unit the value of variable h, while keeping the other variables fixed.

Example: 2016 presidential election results

- Consider a multinomial logistic regression of candidate votes using `Trump` as the baseline category.
- Samples in the data are grouped by county, so each $\mathbf{y}_i$ is a vector with the number of votes per candidate, and $n_i = \sum_{j=1}^{K} y_{ij}$ is the total number of votes by county.
- We consider two models with different predictor variables:
  1. Continuous variable: age groups.
  2. Categorical variable: state.

Example: 2016 presidential election results

The function `vglm` from the package `VGAM` is used to fit multinomial (and ordinal) logistic regression models.

```
library(VGAM)
?vglm
```

vglm {VGAM} R Documentation

Fitting Vector Generalized Linear Models

Description

`vglm` is used to fit vector generalized linear models (VGLMs). This is a very large class of models that includes generalized linear models (GLMs) as a special case.

Usage

```
vglm(formula, family = stop("argument 'family' needs to be assigned"),
     data = list(), weights = NULL, subset = NULL,
     na.action = na.fail, etastart = NULL, mustart = NULL,
     coefstart = NULL, control = vglm.control(...), offset = NULL,
     method = "vglm.fit", model = FALSE, x.arg = TRUE, y.arg = TRUE,
     contrasts = NULL, constraints = NULL, extra = list(),
     form2 = NULL, qr.arg = TRUE, smart = TRUE, ...)
```

Example: 2016 presidential election results

```
> fit <- vglm(formula = cbind(Clinton, Other, Trump) ~ Age_under5 + Age_5_17 + Age_65plus,
              family=multinomial, data=election.data)
> summary(fit)

Pearson residuals:
                       Min     1Q Median      3Q   Max
log(mu[,1]/mu[,3]) -295.9 -40.99 -17.55 -0.4893 481.9
log(mu[,2]/mu[,3]) -109.9 -12.41  -3.67  5.2191 444.7

Coefficients:
                Estimate Std. Error  z value Pr(>|z|)
(Intercept):1  4.714e+00  2.777e-03  1697.82   <2e-16 ***
(Intercept):2 -1.963e-01  6.119e-03   -32.08   <2e-16 ***
Age_under5:1   3.830e-01  3.809e-04  1005.65   <2e-16 ***
Age_under5:2   2.210e-01  8.341e-04   264.96   <2e-16 ***
Age_5_17:1    -2.331e-01  1.199e-04 -1943.91   <2e-16 ***
Age_5_17:2    -9.859e-02  2.594e-04  -380.07   <2e-16 ***
Age_65plus:1  -1.152e-01  6.812e-05 -1690.95   <2e-16 ***
Age_65plus:2  -7.654e-02  1.576e-04  -485.57   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Number of linear predictors:  2

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])

Reference group is level  3  of the response
---
```

Example: continuous variable interpretation

- Consider the variable `Age_65plus`.
- The variable `Age_18_64` is omitted in the data. The interpretation is done with respect to this group.

Example: continuous variable interpretation

- Consider the variable `Age_65plus`.
- The variable `Age_18_64` is omitted in the data. The interpretation is done with respect to this group.
- Increasing the percentage of `Age_65plus` in one unit while keeping the rest of the variables constant implies that `Age_18_64` is decreased in one unit.
- This change will decrease the odds of a vote for `Clinton` compared with a vote for `Trump` by a factor of $exp(-0.15) \approx 0.89$, i.e.,

$$\left(\frac{\mu'_{i1}}{\mu'_{i3}}\right) \approx 0.89 \left(\frac{\mu_{i1}}{\mu_{i3}}\right),$$

where μ'_{i1} and μ'_{i3} indicate the new probabilities after the change on `Age_65plus`.

Example: categorical variable interpretation

By default, the first category of the variable state (`AL`) is the reference group.

```
> fit2 <- vglm(formula = cbind(Clinton, Other, Trump) ~ state_abbr,
               family=multinomial, data=election.data)
> summary(fit2)
                Estimate Std. Error  z value Pr(>|z|)
(Intercept):1  -0.598846   0.001469 -407.675  < 2e-16 ***
(Intercept):2  -3.202201   0.004425 -723.715  < 2e-16 ***
state_abbrAR:1  0.016661   0.002505    6.652 2.89e-11 ***
state_abbrAR:2  0.634093   0.006347   99.897  < 2e-16 ***
state_abbrAZ:1  0.512040   0.002051  249.697  < 2e-16 ***
state_abbrAZ:2  0.931332   0.005481  169.912  < 2e-16 ***
state_abbrCA:1  1.231414   0.001599  770.177  < 2e-16 ***
state_abbrCA:2  1.443476   0.004621  312.405  < 2e-16 ***
...
state_abbrDC:1  3.713440   0.009621  385.984  < 2e-16 ***
state_abbrDC:2  2.894852   0.014961  193.490  < 2e-16 ***
...
state_abbrID:1 -0.165134   0.003144  -52.521  < 2e-16 ***
state_abbrID:2  1.707696   0.005743  297.365  < 2e-16 ***
...
state_abbrKY:1 -0.049812   0.002140  -23.278  < 2e-16 ***
state_abbrKY:2  0.614385   0.005609  109.527  < 2e-16 ***
...
state_abbrMD:1  1.138024   0.001992  571.161  < 2e-16 ***
state_abbrMD:2  1.063731   0.005517  192.813  < 2e-16 ***
...
state_abbrUT:1  0.098787   0.002831   34.890  < 2e-16 ***
state_abbrUT:2  2.643932   0.005065  521.988  < 2e-16 ***
...
...
```

Multivariate exponential family

- Let $\mathbf{Y} = (Y_1, \ldots, Y_q)$ be a random vector. We say that this vector has a distribution in the **multivariate exponential family** if the density (or probability mass function) for $\mathbf{Y} = \mathbf{y} = (y_1, \ldots, y_q)$ is given by

$$f_{\boldsymbol{\theta}}(\mathbf{y}) = \exp\left\{[\mathbf{y}^T\boldsymbol{\theta} - b(\boldsymbol{\theta})]/a(\phi) + c(\mathbf{y}, \phi)\right\},$$

  where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_q)$ and ϕ are parameters, and a, b, c are known functions.

Multivariate exponential family

- Let $\mathbf{Y} = (Y_1, \ldots, Y_q)$ be a random vector. We say that this vector has a distribution in the **multivariate exponential family** if the density (or probability mass function) for $\mathbf{Y} = \mathbf{y} = (y_1, \ldots, y_q)$ is given by

$$f_{\boldsymbol{\theta}}(\mathbf{y}) = \exp\left\{[\mathbf{y}^T\boldsymbol{\theta} - b(\boldsymbol{\theta})]/a(\phi) + c(\mathbf{y}, \phi)\right\},$$

  where $\boldsymbol{\theta} = (\theta_1, \ldots, \theta_q)$ and ϕ are parameters, and a, b, c are known functions.

- **Example:** the multinomial distribution with K classes is in the multivariate exponential family of dimension $q = K - 1$.

Multivariate generalized linear models

- THe multinomial logistic regression model for the data $\{(\mathbf{y}_i, X_i), i = 1, \ldots, n\}$ is a $(K-1)$ dimensional multivariate GLM, in which $\mathbf{y}_i$ has a multinomial distribution. We write $\mathbf{y}_i = (y_{i1}, \ldots, y_{i,K-1})^T$ here.
- For each category $j = 1, \ldots, K-1$, the link function g_j is given by

$$g_j(\boldsymbol{\mu}_i) = \log\left[\frac{\mu_{ij}}{1 - (\mu_{i1} + \cdots + \mu_{i,K-1})}\right] = \log\left[\frac{\mu_{ij}}{\mu_{iK}}\right],$$

  where $\mu_{ij} = \mathbb{E}[\tilde{y}_{ij}]$, with $\tilde{y}_{ij} = \frac{1}{n_i} y_{ij}$ and $n_i = \sum_{j=1}^{K} y_{ij}$.
- The model has $(K-1)p$ different coefficients, and each μ_{ij} is modeled as

$$g_j(\boldsymbol{\mu}_i) = X_i^T \boldsymbol{\beta}_j.$$

Multivariate generalized linear models

- To express all the coefficients in a matrix simultaneously, we write

$$\begin{pmatrix} g_1(\boldsymbol{\mu}_i) \\ g_2(\boldsymbol{\mu}_i) \\ \vdots \\ g_{K-1}(\boldsymbol{\mu}_i) \end{pmatrix} = \mathbf{X}_i\boldsymbol{\beta} = \begin{pmatrix} X_i^T & \mathbf{0} & \cdots & \mathbf{0} \\ \mathbf{0} & X_i^T & \cdots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \cdots & X_i^T \end{pmatrix} \begin{pmatrix} \boldsymbol{\beta}_1 \\ \boldsymbol{\beta}_2 \\ \vdots \\ \boldsymbol{\beta}_{K-1} \end{pmatrix}.$$

  Each $\mathbf{X}_i$ is a matrix of size $(K-1) \times (K-1)p$, and $\boldsymbol{\beta} \in \mathbb{R}^{(K-1)p}$.
- The data matrix for the regression is $\mathbf{X} = (\mathbf{X}_1^T, \cdots, \mathbf{X}_n^T)^T$, a matrix with $(K-1)n$ rows and $(K-1)p$ columns.

Inference for the multinomial logistic model

- Model fitting is done with the maximum likelihood estimator, which is calculated numerically.
- As usual, the MLE estimators $\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_{K-1}$ have a large-sample normal distribution.
- Statistical inference can use the Wald, likelihood ratio or score tests.
- As in logistic regression, the Wald test might exhibit some aberrant behavior in which the test statistic is not monotonically increasing as the effect size increases. This is called the *Hauck-Donner effect*.

Deviance

- The derivation of the deviance for binomial GLMs generalizes to multinomial GLMs.
- The deviance and the Pearson statistic are

$$D = 2\sum_{i=1}^{n}\sum_{j=1}^{K} n_i\tilde{y}_{ij}\log\left[\frac{n_i\tilde{y}_{ij}}{n_i\hat{\mu}_{ij}}\right],$$

$$X^2 = 2\sum_{i=1}^{n}\sum_{j=1}^{K}\frac{(n_i\tilde{y}_{ij} - n_i\hat{\mu}_{ij})^2}{n_i\hat{\mu}_{ij}}.$$

- For grouped data with each $n_i \to \infty$ and non-negligible probabilities, D and X^2 have an asymptotic chi-squared distribution with $(K-1)n - (K-1)p$ degrees of freedom.

Inference for the multinomial logistic model

- The covariance matrix of $\hat{\boldsymbol{\beta}} = (\hat{\boldsymbol{\beta}}_1, \ldots, \hat{\boldsymbol{\beta}}_{K-1})$ has size $(K-1)p \times (K-1)p$.
- By differentiating the likelihood, the entries of the expected information for two variables within the same class j are

$$-\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \beta_{jk} \partial \beta_{jk'}} = \mathbf{X}^T \mathbf{W}_{(j)} \mathbf{X},$$

and for different classes j and j',

$$-\frac{\partial^2 \ell(\boldsymbol{\beta})}{\partial \beta_{jk} \partial \beta_{j'k'}} = \mathbf{X}^T \mathbf{W}_{(j,j')} \mathbf{X},$$

with $\mathbf{W}_{(j)}$ and $\mathbf{W}_{(jj')}$ diagonal matrices such that

$$(\mathbf{W}_{(j)})_{ii} = \mu_{ij}(1-\mu_{ij}),$$

$$(\mathbf{W}_{(jj')})_{ii} = \mu_{ij}\mu_{ij'}.$$

Example: model inference

```
> fit <- vglm(formula = cbind(Clinton, Other, Trump) ~ . - state_abbr,
              family=multinomial, data=election.data)

Pearson residuals:
                       Min     1Q Median    3Q   Max
log(mu[,1]/mu[,3]) -274.3 -17.35 -5.601 9.136 285.0
log(mu[,2]/mu[,3]) -113.4  -6.19 -1.116 5.146 292.3

Coefficients:
                  Estimate Std. Error z value Pr(>|z|)
(Intercept):1    -2.235e+00  1.328e-02 -168.29   <2e-16 ***
(Intercept):2    -3.597e+00  3.039e-02 -118.37   <2e-16 ***
Age_under5:1      2.466e-02  5.455e-04   45.20   <2e-16 ***
Age_under5:2      1.524e-01  1.158e-03  131.62   <2e-16 ***
Age_5_17:1       -5.829e-02  1.987e-04 -293.29   <2e-16 ***
Age_5_17:2       -6.463e-03  4.314e-04  -14.98   <2e-16 ***
Age_65plus:1     -2.839e-02  9.791e-05 -289.92   <2e-16 ***
Age_65plus:2     -1.163e-02  2.237e-04  -51.99   <2e-16 ***
Female:1          5.955e-02  2.078e-04  286.61   <2e-16 ***
Female:2         -2.845e-02  4.605e-04  -61.77   <2e-16 ***
White:1          -1.125e-02  3.413e-05 -329.77   <2e-16 ***
White:2          -8.888e-03  6.518e-05 -136.37   <2e-16 ***
Black:1           3.272e-03  3.679e-05   88.95   <2e-16 ***
Black:2          -2.345e-02  7.627e-05 -307.50   <2e-16 ***
Foreignborn:1     2.500e-02  3.758e-05  665.35   <2e-16 ***
Foreignborn:2     8.443e-03  9.189e-05   91.88   <2e-16 ***
Edu_highschool:1  1.475e-02  7.938e-05  185.81   <2e-16 ***
Edu_highschool:2  4.823e-02  1.849e-04  260.89   <2e-16 ***
Edu_bachelor:1    1.038e-02  5.675e-05  182.94   <2e-16 ***
Edu_bachelor:2    1.881e-02  1.215e-04  154.88   <2e-16 ***
```

Example: model inference

```
Home_ownership:1 -1.488e-02  4.689e-05 -317.34   <2e-16 ***
Home_ownership:2 -1.940e-02  1.068e-04 -181.59   <2e-16 ***
Income:1          1.599e-05  8.822e-08  181.20   <2e-16 ***
Income:2         -1.834e-05  1.924e-07  -95.33   <2e-16 ***
Poverty:1         2.198e-02  9.063e-05  242.53   <2e-16 ***
Poverty:2         8.376e-03  2.029e-04   41.27   <2e-16 ***
Density:1        -2.908e-06  5.963e-08  -48.77   <2e-16 ***
Density:2         6.156e-06  1.250e-07   49.25   <2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Number of linear predictors:  2

Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])

Residual deviance: 4646991 on 6196 degrees of freedom

Log-likelihood: -2350855 on 6196 degrees of freedom

Number of iterations: 4

Warning: Hauck-Donner effect detected in the following estimate(s):
'(Intercept):2'

Reference group is level  3  of the response
```

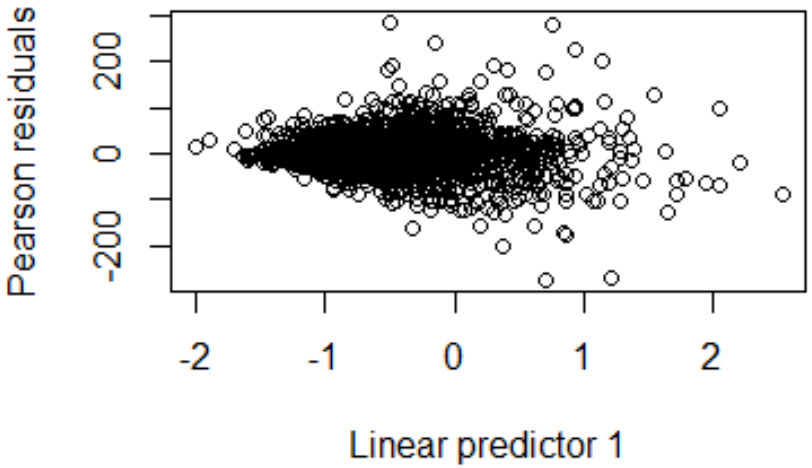
Model diagnostics

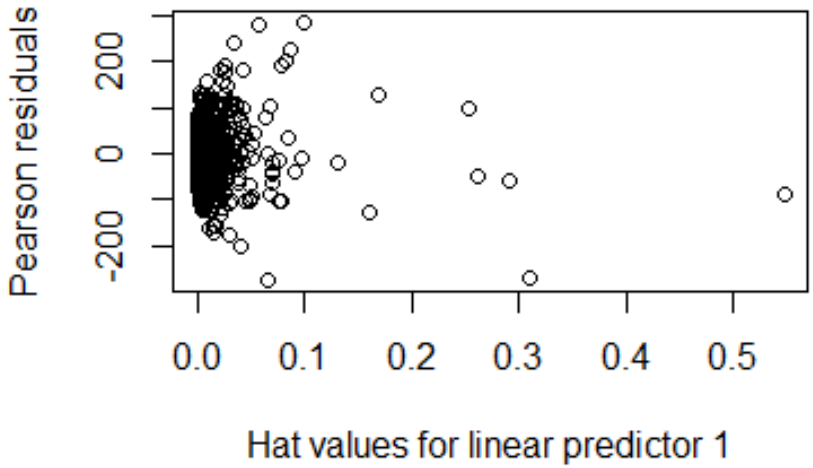
- For grouped data, the residuals (either Pearson, deviance or standardized) are approximately distributed as standard normal. This can be checked with a normal quantile-quantile (q-q) plot.
- Model misspecification can be detected by plotting the residuals against the explanatory variables, leverage, or the linear predictors:
  - **Mean misspecification.** Trends in the mean might indicate that the link function is incorrect, some variables are missing, or variable transformations are needed to improve the fit.
  - **Variance misspecification.** Non-unit variance indicates that the dispersion parameter is not correctly specified. Non-constant variance suggests that the random component of the GLM is incorrect. In those situations, consider using a different response distribution or a quasi-likelihood.
  - **Outliers.** Observations that have a significantly large residual value may be removed to improve model fit.

Model diagnostics

- For `vglm`, model checking with the Pearson residuals can be performed using the `plot` function of the fitted model.
- The function shows the Pearson residuals against the linear predictors and the hat values (i.e., the leverage).

```
> plot(fit)
```





Model checking: Q-Q plot

```
# Compute standardized residuals
fit.stdres <- residuals(fit, "stdres")
qqnorm(fit.stdres[,1],  ylab="Standardized Residuals 1",
       xlab="Normal Scores")
qqline(fit.stdres[,1])
```

Model inference

- The (scaled) deviance can be used as a test statistic for the test of comparing the fitted model with the saturated model.

```
> 1 - pchisq(deviance(fit), df = fit@df.residual)
0
```

Model inference

- The (scaled) deviance can be used as a test statistic for the test of comparing the fitted model with the saturated model.

```
> 1 - pchisq(deviance(fit), df = fit@df.residual)
0
```

- We can also compare the fit of two different models using their (scaled) deviance differences.

```
> fit2 <- vglm(formula = cbind(Clinton, Other, Trump) ~ . - state_abbr -
Density, family=multinomial, data=election.data)

> anova.vglm(fit2, fit, type = 1)

Analysis of Deviance Table

Model 1: cbind(Clinton, Other, Trump) ~ . - state_abbr - Density
Model 2: cbind(Clinton, Other, Trump) ~ . - state_abbr
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      6198    4654319
2      6196    4646991  2   7327.6 < 2.2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Model inference

- The (scaled) deviance can be used as a test statistic for the test of comparing the fitted model with the saturated model.

```
> 1 - pchisq(deviance(fit), df = fit@df.residual)
0
```

- We can also compare the fit of two different models using their (scaled) deviance differences.

```
> fit2 <- vglm(formula = cbind(Clinton, Other, Trump) ~ . - state_abbr -
Density, family=multinomial, data=election.data)

> anova.vglm(fit2, fit, type = 1)

Analysis of Deviance Table

Model 1: cbind(Clinton, Other, Trump) ~ . - state_abbr - Density
Model 2: cbind(Clinton, Other, Trump) ~ . - state_abbr
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1      6198    4654319
2      6196    4646991  2   7327.6 < 2.2e-16 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Problems with the p-values

- When the sample size is very large, hypothesis tests will usually choose the more complex model.
- “All models are wrong”: in practice, the null hypothesis is wrong.
- Variables are usually at least slightly correlated with the response. Hence, the p-value can be made arbitrarily small if the sample size is large enough.
- A linear model is almost never correctly specified, and with enough sample size the model will be rejected in favor of the more complex model .
- Additionally, the choice of the significance level (such as 0.05) is arbitrary.
- These remarks suggest that the p-values should be used with caution. In practice, model parsimony is usually preferred, and hence variables with small effects might be discarded even if they are significant.

Problems with the p-values

Some possible solutions:

- Use a quasi-likelihood model to adjust the dispersion parameter and improve model fit. Unfortunately, the quasi-multinomial is not implemented in R, but quasi-binomial can be used for each pair of categories.
- Compare the magnitude of the coefficients or the test statistics (z-values).
- Compare the difference in deviance when variables are added or dropped from the model.

Comparing coefficient estimates

- Comparing the magnitude of the coefficients requires that the variables have the same scale.
- Linear transformations (such as centering and scaling) do not affect the fit of a GLM if an intercept is included. If $\tilde{\mathbf{X}}$ is the data matrix after centering and scaling each variable, then

$$\mathbf{X}\boldsymbol{\beta} = \tilde{\mathbf{X}}\tilde{\boldsymbol{\beta}} + \tilde{\beta}_0$$

for some $\tilde{\boldsymbol{\beta}}$ and $\tilde{\beta}_0$. Hence, the fitted values $\hat{\mu}_i$ do not change.

Example: model inference

```
scaled.data <- election.data
scaled.data[, -(1:4)] <- scale(election.data[, -(1:4)])

fit.scaled <- vglm(formula = cbind(Clinton, Other, Trump) ~ . - state_abbr,
                   family=multinomial, data=scaled.data)
summary(fit.scaled)
```

Example: model inference

```
Coefficients:
                   Estimate Std. Error  z value     Pr(>|z|)
(Intercept):1    -0.6287091  0.0003184 -1974.49   <2e-16 ***
(Intercept):2    -2.6558394  0.0007578 -3504.88   <2e-16 ***
Age_under5:1      0.0291720  0.0006454    45.20   <2e-16 ***
Age_under5:2      0.1802504  0.0013695   131.62   <2e-16 ***
Age_5_17:1       -0.1943116  0.0006625  -293.29   <2e-16 ***
Age_5_17:2       -0.0215461  0.0014381   -14.98   <2e-16 ***
Age_65plus:1     -0.1245376  0.0004296  -289.92   <2e-16 ***
Age_65plus:2     -0.0510183  0.0009813   -51.99   <2e-16 ***
Female:1          0.1313235  0.0004582   286.61   <2e-16 ***
Female:2         -0.0627315  0.0010156   -61.77   <2e-16 ***
White:1          -0.1776851  0.0005388  -329.77   <2e-16 ***
White:2          -0.1403299  0.0010291  -136.37   <2e-16 ***
Black:1           0.0473951  0.0005328    88.95   <2e-16 ***
Black:2          -0.3396495  0.0011046  -307.50   <2e-16 ***
Foreignborn:1     0.1380415  0.0002075   665.35   <2e-16 ***
Foreignborn:2     0.0466130  0.0005073    91.88   <2e-16 ***
Edu_highschool:1  0.1019497  0.0005487   185.81   <2e-16 ***
Edu_highschool:2  0.3334017  0.0012780   260.89   <2e-16 ***
Edu_bachelor:1    0.0916717  0.0005011   182.94   <2e-16 ***
Edu_bachelor:2    0.1661314  0.0010727   154.88   <2e-16 ***
Home_ownership:1 -0.1171282  0.0003691  -317.34   <2e-16 ***
Home_ownership:2 -0.1526577  0.0008407  -181.59   <2e-16 ***
Income:1          0.0885796  0.0004888   181.20   <2e-16 ***
Income:2         -0.1016041  0.0010659   -95.33   <2e-16 ***
Poverty:1         0.1426476  0.0005882   242.53   <2e-16 ***
Poverty:2         0.0543576  0.0013170    41.27   <2e-16 ***
Density:1        -0.0050390  0.0001033   -48.77   <2e-16 ***
Density:2         0.0106649  0.0002165    49.25   <2e-16 ***
```

Multiclass classification

- The multinomial logistic regression is commonly used as a multi-class classifier.
- The class label for each observation is estimated as the class with the largest fitted probability, that is

$$\hat{y}_i = \underset{j \in \{1,\ldots,K\}}{\operatorname{argmax}} \ \hat{\mu}_{ij}.$$

  This is equivalent to choosing a cutoff of 0.5 in logistic regression for two classes.
- The expected error can be estimated via sample splitting or cross-validation.
- Using different cutoffs for each class is problematic because it is not guaranteed that exactly one class will be selected.

Ordinal responses

Setting

- Here we consider data for which the response is an ordinal variable with C classes.
- Let y_i denote the response outcome category for subject i, and $\mathbf{y}_i = (y_{i1}, \ldots, y_{iC})$ its vector representation, such that $y_i = j$ means that $y_{ij} = 1$ and $y_{ik} = 0$ for $k \neq j$.
- We model the likelihood that y_i is in a certain class j given X_i.

Cumulative logit model

- For each $j = 1, \ldots, C$, denote by π_{ij} to the probability that $y_i = j$ given X_i, i.e.,

$$\pi_{ij} = \mathbb{P}(y_i = j | X_i).$$

- The model can be expressed in terms of cumulative probabilities as

$$\mathbb{P}(y_i \leq j | X_i) = \pi_{i1} + \cdots + \pi_{ij}.$$

- For $j = 1, \ldots, C-1$, the **cumulative logits** are the logits of the cumulative probabilities

$$\begin{aligned} \text{logit}\left[\mathbb{P}(y_i \leq j | X_i)\right] &= \log\left(\frac{\mathbb{P}(y_i \leq j)}{\mathbb{P}(y_i > j)}\right) \\ &= \log\left(\frac{\sum_{h=1}^{j} \pi_{ih}}{\sum_{h=j+1}^{C} \pi_{ih}}\right). \end{aligned}$$

Cumulative logit model

- To model all the cumulative probabilities simultaneously, the **cumulative logit model** uses a common intercept for all the logits with an individual incercept for each class

$$\text{logit}\left[\mathbb{P}(y_i \leq j | X_i)\right] = \alpha_j + X_i^T \boldsymbol{\beta}.$$

  We assume here that the intercept column is excluded from $\mathbf{X}$.
- The intercepts are strictly increasing in j because the cumulative probabilities are strictly increasing.

$$\alpha_1 < \alpha_2 < \cdots < \alpha_{C-1}.$$

Model interpretation

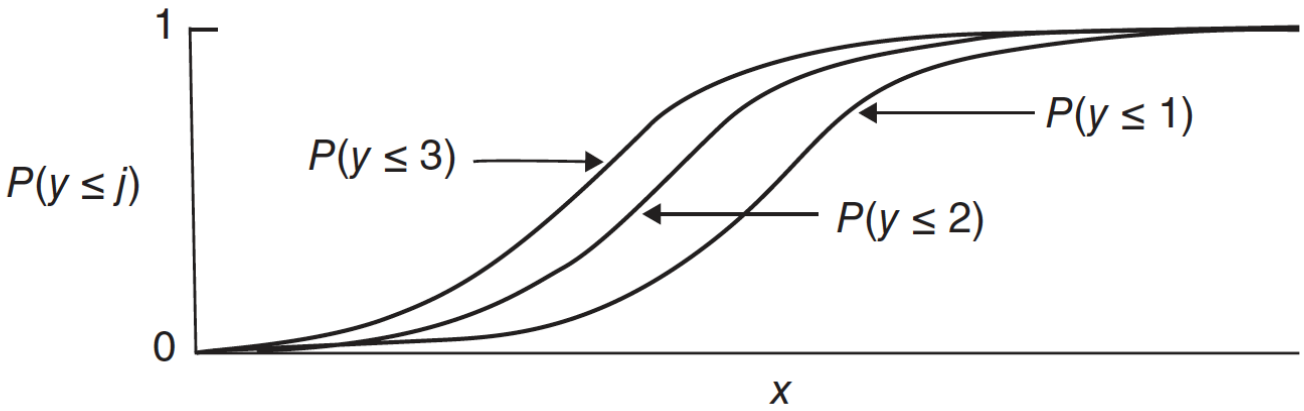
- The coefficient $\boldsymbol{\beta}$ has the same interpretation as the previous models: a change in the predictors from X_i to $\tilde{X}_i$ will result in a proportional change to the cumulative odds

$$\frac{\mathbb{P}(y_i \leq j|\tilde{X}_i)}{1-\mathbb{P}(y_i \leq j|\tilde{X}_i)} = \exp\left[(\tilde{X}_i - X_i)^T\boldsymbol{\beta}\right]\left(\frac{\mathbb{P}(y_i \leq j|X_i)}{1-\mathbb{P}(y_i \leq j|X_i)}\right).$$

- Hence, increasing one unit of X_{ih} while keeping all the other variables fixed will result in a proportional change to the odds by a factor of $\exp(\beta_h)$. This imply that positive coefficients indicate a preference for smaller values of j.
- Similarly, the cumulative odds of classes j and k are related by a factor of $\exp(\alpha_j - \alpha_k)$.
- When the covariates are categorical, α_j is the cumulative logit j for the reference group.

Model interpretation

- **Proportional odds:** the same proportionality constant applies to all the cumulative logits, implying some "parallelism" of the functions.



Model interpretation

- For $j = 1, \ldots, C-1$,the cumulative probabilities are given by

$$\mathbb{P}(y_i \leq j | X_i) = \frac{\exp(\alpha_j)\exp(X_i^T \boldsymbol{\beta})}{1 + \exp(\alpha_j)\exp(X_i^T \boldsymbol{\beta})}.$$

This formula implies that the probabilities are *stochastically order*: for all X_i, $\tilde{X}_i$ either

$$\mathbb{P}(y_i \leq j | X_i) \leq \mathbb{P}(y_i \leq j | \tilde{X}_i) \quad \text{for all } j = 1, \ldots, C,$$

or

$$\mathbb{P}(y_i \leq j | X_i) \geq \mathbb{P}(y_i \leq j | \tilde{X}_i) \quad \text{for all } j = 1, \ldots, C.$$

Model interpretation

- For $j = 1, \ldots, C-1$,the cumulative probabilities are given by

$$\mathbb{P}(y_i \leq j | X_i) = \frac{\exp(\alpha_j)\exp(X_i^T \boldsymbol{\beta})}{1 + \exp(\alpha_j)\exp(X_i^T \boldsymbol{\beta})}.$$

This formula implies that the probabilities are *stochastically order*: for all X_i, $\tilde{X}_i$ either

$$\mathbb{P}(y_i \leq j | X_i) \leq \mathbb{P}(y_i \leq j | \tilde{X}_i) \quad \text{for all } j = 1, \ldots, C,$$

or

$$\mathbb{P}(y_i \leq j | X_i) \geq \mathbb{P}(y_i \leq j | \tilde{X}_i) \quad \text{for all } j = 1, \ldots, C.$$

- The class probabilities for $j = 2, \ldots, C$ are

$$\mathbb{P}(y_i = j | X_i) = \frac{\exp(\alpha_j)\exp(X_i^T \boldsymbol{\beta})}{1 + \exp(\alpha_j)\exp(X_i^T \boldsymbol{\beta})} - \frac{\exp(\alpha_{j-1})\exp(X_i^T \boldsymbol{\beta})}{1 + \exp(\alpha_{j-1})\exp(X_i^T \boldsymbol{\beta})}.$$

Model likelihood

- The log-likelihood of the cumulative logit model can be expressed using the multinomial log-likelihood function

$$\ell(\alpha_1, \ldots, \alpha_{C-1}, \boldsymbol{\beta}) = \sum_{i=1}^{n} \sum_{j=1}^{C} y_{ij} \log(\pi_{ij}).$$

- The expected value of the response is

$$\mathbb{E}[\mathbf{y}_i] = \boldsymbol{\pi}_i = (\pi_{i1}, \ldots, \pi_{iC}).$$

The multivariate link function is $g(\pi) = (g_1(\boldsymbol{\pi}), \ldots, g_{C-1}(\boldsymbol{\pi}))^T$, where

$$g_j(\boldsymbol{\pi}_i) = \log\left(\frac{\sum_{h=1}^{j} \pi_{ij}}{1 - \sum_{h=1}^{j} \pi_{ij}}\right) = \text{logit}\left[\mathbb{P}(y_i \leq j | X_i)\right].$$

- Observe that the link function of the cumulative logit model is not the canonical link function.

Ordinal regression as a multivariate GLM

- To express all the coefficients in a matrix simultaneously, we write

$$\begin{pmatrix} g_1(\boldsymbol{\pi}_i) \\ g_2(\boldsymbol{\pi}_i) \\ \vdots \\ g_{C-1}(\boldsymbol{\pi}_i) \end{pmatrix} = \mathbf{X}_i\tilde{\boldsymbol{\beta}} = \begin{pmatrix} 1 & 0 & \cdots & 0 & X_i^T \\ 0 & 1 & \cdots & 0 & X_i^T \\ \vdots & \vdots & \ddots & & \vdots \\ \vdots & \vdots & \ddots & & \vdots \\ 0 & 0 & \cdots & 1 & X_i^T \end{pmatrix} \begin{pmatrix} \alpha_1 \\ \alpha_2 \\ \vdots \\ \alpha_{C-1} \\ \boldsymbol{\beta} \end{pmatrix}.$$

  Each $\mathbf{X}_i$ is a matrix of size $(C-1) \times (C-1+p)$, and $\tilde{\boldsymbol{\beta}} \in \mathbb{R}^{(C-1+p)}$.

- The data matrix for the regression $\mathbf{X} = (\mathbf{X}_1^T, \cdots, \mathbf{X}_n^T)^T$ has $(C-1)n$ rows and $(C-1+p)$ columns.

Inference and model checking

- The MLE of the parameters is solved numerically.
- Inference, model checking and model comparison are performed in a similar manner to the multinomial logistic model, using the asymptotic normality of $\hat{\boldsymbol{\beta}}$.
- The covariance matrix of $\hat{\boldsymbol{\beta}}$ is obtained using the chain rule.
- The asymptotic distribution of several quantities of interest (scaled deviance, residuals, Pearson statistic, etc.) is only guaranteed for grouped data.

Alternative models

- A more flexible model uses separate coefficients $\{\boldsymbol{\beta}_j\}$ for each cumulative logit.
- This model can fit the data better due to the increased number of parameters, but interpretation is harder, especially when C is large.
- Additionally, the estimated curves are not always parallel, which violates the proper order of the cumulative probabilities.
- The two models (common and separate coefficient) can be compared with a likelihood ratio test (or deviance difference).

Example: WVS data

- The World Values Survey (WVS) dataset (available on `carData` R package) contains a sample of responses to surveys made in Australia, Norway, Sweden, and the United States around 1995-1997.
- Each person was asked the question *"Do you think that what the government is doing for people in poverty in this country is too little, about the right amount or too much?*
- The data includes the following covariates
  - `religion:` binary.
  - `degree:` binary.
  - `country:` categorical.
  - `age:` continuous.
  - `gender:` binary

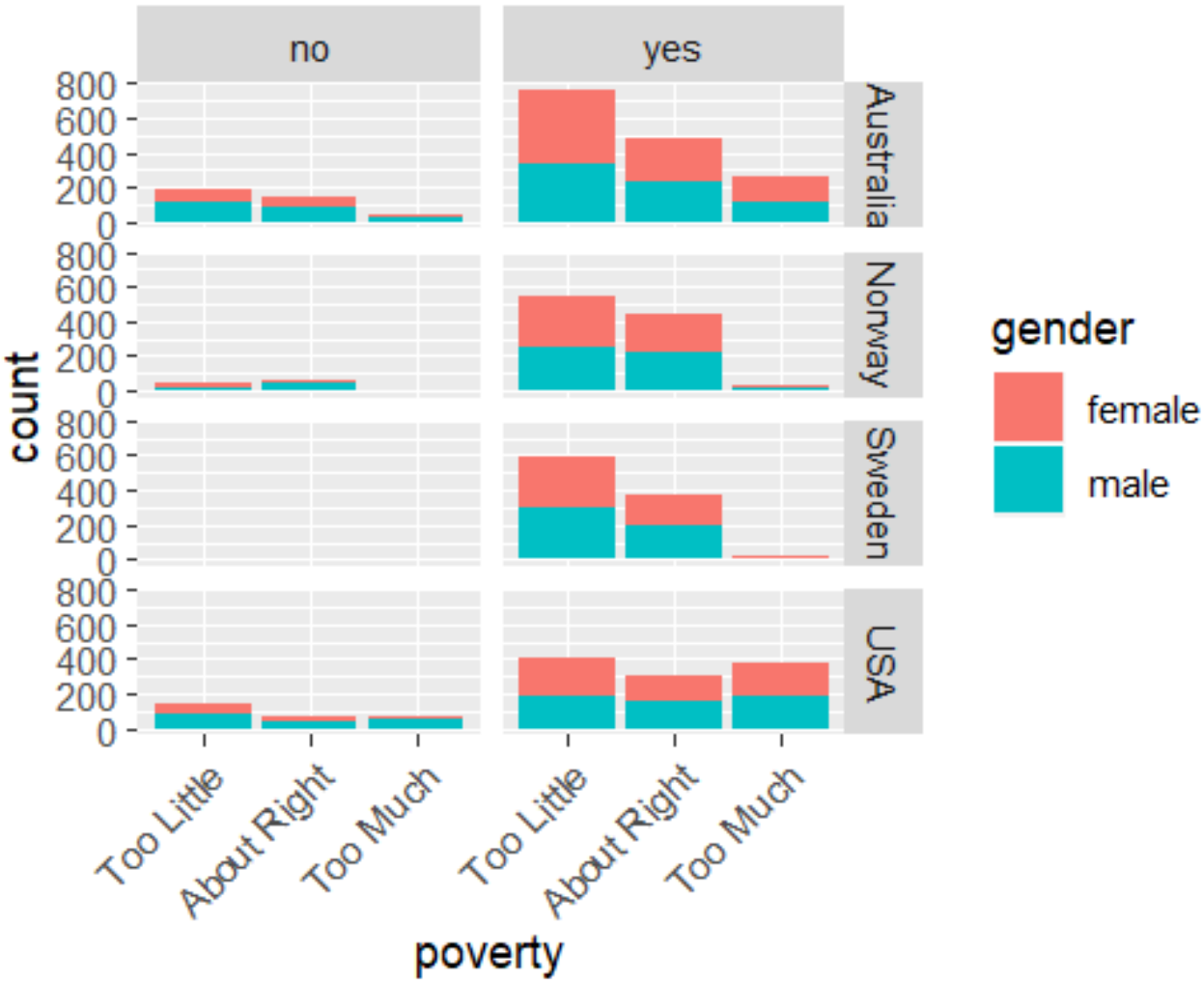
Example: WVS data

```
> library(carData)
> data(WVS)
> head(WVS)

      poverty religion degree country age gender
1  Too Little      yes     no     USA  44   male
2 About Right      yes     no     USA  40 female
3  Too Little      yes     no     USA  36 female
4    Too Much      yes    yes     USA  25 female
5  Too Little      yes    yes     USA  39   male
6 About Right      yes     no     USA  80 female


> # Analyze the distribution using a bar plot
> library(ggplot2)
> ggplot(WVS, aes(x = poverty, fill = gender)) +
        geom_bar() +
        facet_grid(country ~ religion, margins = FALSE) +
        theme(axis.text.x = element_text(angle = 45, hjust = 1, vjust = 1))
```

Example: WVS data



Model fitting

```
> library(VGAM)
> WVS.reg <- vglm(formula = poverty ~ ., family = cumulative(parallel = TRUE),
                  data = WVS)
> summary(WVS.reg)

Pearson residuals:
Min      1Q Median     3Q    Max
logit(P[Y<=1]) -1.968 -1.0881 0.6046 0.9040 1.8792
logit(P[Y<=2]) -3.838  0.2055 0.2545 0.5674 0.9634

Coefficients:
               Estimate Std. Error z value Pr(>|z|)
(Intercept):1  0.729769   0.104044   7.014 2.32e-12 ***
(Intercept):2  2.532483   0.110154  22.990  < 2e-16 ***
religionyes   -0.179733   0.076565  -2.347 0.018902 *
degreeyes     -0.140918   0.066714  -2.112 0.034663 *
countryNorway  0.322353   0.075444   4.273 1.93e-05 ***
countrySweden  0.603300   0.080895   7.458 8.79e-14 ***
countryUSA    -0.617773   0.068391  -9.033  < 2e-16 ***
age           -0.011141   0.001557  -7.157 8.24e-13 ***
gendermale    -0.176370   0.052877  -3.335 0.000851 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Model fitting

```
Number of linear predictors:  2

Names of linear predictors: logit(P[Y<=1]), logit(P[Y<=2])

Residual deviance: 10402.59 on 10753 degrees of freedom

Log-likelihood: -5201.296 on 10753 degrees of freedom

Number of iterations: 5

No Hauck-Donner effect found in any of the estimates

Exponentiated coefficients:
religionyes     degreeyes countryNorway countrySweden   countryUSA          age
0.8354929     0.8685603     1.3803723     1.8281418     0.5391437     0.9889208
gendermale
0.8383078
```

Model interpretation

- A religious person, a male, or a person with a degree is more likely to think that the government is doing too much for people in poverty, compared to an individual of the opposite category with the same characteristics
- Individuals from Norway or Sweden are more likely to think that the government is doing too little than individuals from Australia with the same characteristics, while USA individuals are more likely to answer ”too much”.
- The likelihood of answering “too much” increases with age compared to a young person that has the same characteristics.

Interaction terms

- The bar plot suggested that there is a difference between religious and non-religious people, most notably in the USA samples.
- This effect can be modeled using an interaction.

```
> WVS.reg2 <- vglm(formula = poverty ~ . + religion*country,
                   family = cumulative(parallel = TRUE), data = WVS)
> summary(WVS.reg2)
```

Interaction terms

```
Coefficients:
Estimate Std. Error z value Pr(>|z|)
(Intercept):1              0.637977   0.121953   5.231 1.68e-07 ***
(Intercept):2              2.447521   0.126973  19.276  < 2e-16 ***
religionyes               -0.065228   0.111507  -0.585 0.558569
degreeyes                 -0.140326   0.066807  -2.100 0.035688 *
countryNorway              0.016095   0.209399   0.077 0.938732
countrySweden             -0.299783   0.496044  -0.604 0.545612
countryUSA                -0.180385   0.149744  -1.205 0.228348
age                       -0.011138   0.001559  -7.147 8.89e-13 ***
gendermale                -0.178468   0.052932  -3.372 0.000747 ***
religionyes:countryNorway  0.326680   0.223411   1.462 0.143676
religionyes:countrySweden  0.894708   0.503067   1.779 0.075321 .
religionyes:countryUSA    -0.551763   0.167073  -3.303 0.000958 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1

Number of linear predictors:  2
```

Model comparison

The models can be compared with the likelihood ratio test (or equivalently, the scaled deviance differences).

```
> anova.vglm(WVS.reg, WVS.reg2, type = 1)

Analysis of Deviance Table

Model 1: poverty ~ .
Model 2: poverty ~ . + religion * country
  Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
1     10753      10403
2     10750      10381  3   21.665 7.658e-05 ***
---
Signif. codes:  0 *** 0.001 ** 0.01 * 0.05 . 0.1   1
```

Ordinal class prediction

- The class labels of the ordinal responses can be estimated in a similar way to the multinomial logistic regression

$$\hat{y}_i = \underset{j \in \{1,\ldots,C\}}{\operatorname{argmax}} \ \hat{\pi}_{ij}.$$

- Alternatively, the classes can be estimated using a *one vs. the rest* approach:
  1. Using the fitted linear prediction $\mathbf{X}\hat{\boldsymbol{\beta}}$, start by obtaining a classifier for class C vs classes $\{1, \ldots, C-1\}$ (or class 1 vs. the rest). An optimal cutoff can be estimated using a grid of values.
  2. Repeat this process for the remaining classes, until all classes have been classified.
- The expected error can be estimated using sample splitting or cross-validation.

References

- A. Agresti, *Foundations of Linear and Generalized Linear Models*, Wiley (2013). Chapters 5 and 6.