STAT 528 - Advanced Regression Analysis II

Binary response regression (part I)

Daniel J. Eck
Department of Statistics
University of Illinois

Last time

- Optimization
- Sufficiency
- Maximum Entropy
- Wrap-up

Learning Objectives Today

- logistic regression
- data analysis
- connecting theory to application

Background

We suppose that we have a sample of data (y_i, x_i), $i = 1, \ldots, n$ where

- y_i is a scalar response variable
- x_i is a vector of predictors.

Recall from the exponential family notes that the log likelihood of the exponential family is of the form

$$l(\theta) = \langle y, \theta \rangle - c(\theta), \tag{1}$$

where

- $y \in \mathbb{R}^n$ is a vector statistic having components
- $\theta \in \mathbb{R}^n$ is the canonical parameter vector.

In those notes θ is unconstrained and the likelihood (1) corresponds to a saturated regression model, one parameter for every observation.

A canonical linear submodel of an exponential family is a submodel having parameterization

$$\theta = M\beta,$$

and log likelihood

$$l(\beta) = \langle M'y, \beta \rangle - c(M\beta). \tag{2}$$

In an exponential family GLM, the saturated model canonical parameter vector θ is "linked'' to the saturated model mean value parameter vector through the change-of-parameter mappings $g(\theta)$.

We can write

$$\mu = \mathsf{E}_\theta(Y) = g(M\beta)$$

which implies that we can write

$$g^{-1}\left(\mathsf{E}_\theta(Y)\right) = M\beta.$$

This is the basis of exponential family generalized linear models with link function g^{-1}.

Logistic regression model

The logistic regression model is one of the most widely used and studied GLMs in practice.

It is [one of] the most important model for binary response data, being commonly used for a wide variety of applications.

The logistic regression model is used for analyzing a binary response variable, $y_i \in \{0, 1\}$ where a 1 encodes a "success" and a 0 encodes a "failure."

The logistic regression model allows for users to model the probability of success as a function of covariates.

For a binary response variable Y and a vector of predictors X, let

$$\pi(x) = P(Y = 1|X = x).$$

The logistic regression model is then

$$\pi(x) = P(Y = 1|X = x) = \frac{\exp(x'\beta)}{1 + \exp(x'\beta)}. \tag{3}$$

Equivalently, the *logit* (log-odds) of the response variable has a linear relationship in the canonical submodel parameters:

$$\text{logit}(\pi(x)) = \log\left(\frac{\pi(x)}{1 - \pi(x)}\right) = x^T\beta.$$

In vector notation, we can express the above as

$$\boldsymbol{\pi} = \frac{\exp(M\beta)}{1 + \exp(M\beta)} = \frac{1}{1 + \exp(-M\beta)} \quad \text{and} \quad \text{logit}(\boldsymbol{\pi}) = M\beta$$

where the above $\exp(\cdot)$ and $\text{logit}(\cdot)$ operations are understood as componentwise operations.

We will again connect the log likelihood of independent Bernoulli trials to canonical linear submodels

$$\begin{aligned}
&\sum_{i=1}^{n} y_i \log(\pi(x_i)) + (1 - y_i) \log(1 - \pi(x_i)) \\
&\qquad = \sum_{i=1}^{n} y_i \log\left(\frac{\pi(x_i)}{1 - \pi(x_i)}\right) - \log(1 - \pi(x_i)) \\
&\qquad = \sum_{i=1}^{n} y_i x_i' \beta - \log(1 + \exp(x_i' \beta)) \\
&\qquad = \langle M' y, \beta \rangle - c_\beta(\beta),
\end{aligned}$$

where M has rows x_i'.

Takeaways

- The logistic regression model is an exponential family model whose log likelihood can be written in canonical form. As such, the nice properties discussed over the last four lectures hold for this model.
- Note the differences between logistic and linear regression: the logistic regression model does not possess an additive error structure (ie signal plus noise); the change of parameters map g is not the identity function; the mean-value parameter is a success probability
- In the first point above, it is interesting to note how John A Nelder, one of the creators of GLMs, identifies the statistics of his day as too focused on mathematical properties of error distributions instead of studying the mechanisms of signal which is of interest to scientists and technologists (See Section 2 of this paper).

Data Analysis

- We will now apply logistic regression on a data set
- We will teach some basic data wrangling steps using `dplyr` within `tidyverse`

The dplyr package within tidyverse

```
#install.packages("tidyverse")
library(tidyverse)
```

dplyr provides comprehensive tools for data manipulations (or wrangling). The five main "verbs" are:

- `select()`: choose from a subset of the columns
- `filter()`: choose a subset of the rows based on logical criteria
- `arrange()`: sort the rows based on values of the columns.
- `mutate()`: add or modify the definitions of the column, and create columns that are functions of existing columns.
- `summarise()`: collapse a data frame down to a single row (per group) by aggregating vectors into a single value. Often used in conjunction with `group_by()`

The pipe operator

The pipe operator %>% allows for verbs to be strung in succession so that complicated manipulations can be combined within a single easily digestible sentence.

```
data %>%
    inner_function() %>%
    outer_function()
```

Example: CCSO data

We will demonstrate logistic regression modeling on the Champaign County Sheriff's Office (CCSO) data frame.

We now load in the CCSO data frame using the fread (**f**ast **read**) function from the `data.table` package (can also use read_csv in the `tidyverse`) and perform most of our data wrangling operations using `dplyr`.

```
rm(list = ls())
library(tidyverse)
library(data.table)
```

Let's load in and wrangle the data.

```
## load in data
system.time(CCSO <- fread("https://uofi.box.com/shared/static/9elozjsg99bgcb7gb546wlfr3r2gc9b7.csv"))
```

```
##    user  system elapsed
##   0.303   0.095   5.227
```

```
dim(CCSO)
```

```
## [1] 67764    35
```

```
## data wrangling
CCSO_small <- CCSO %>% rename(Days = "Days in Jail", Age = "Age at Arrest",
                          Date = "BOOKING DATE", Sex = "SEX", Race = "RACE",
                          Crime = "CRIME CODE", Agency = "ARREST AGENCY") %>%
  mutate(atleastone = ifelse(Days > 0,1,0)) %>%
  filter(Crime == "OTHER TRAFFIC OFFENSES") %>%
  filter(Race %in% c("Asian/Pacific Islander","Black","White","Hispanic")) %>%
  filter(Sex %in% c("Female","Male")) %>%
  dplyr::select(atleastone, Age, Sex, Date, Race) %>%
  mutate(Race = fct_drop(Race), Sex = fct_drop(Sex))
CCSO_small <- CCSO_small[complete.cases(CCSO_small), ]

head(CCSO_small, 5)
```

```
##    atleastone Age    Sex     Date     Race
## 1:          0  22   Male 1/1/2011    White
## 2:          0  26   Male 1/1/2011    White
## 3:          0  32 Female 1/1/2011    White
## 4:          0  22   Male 1/2/2011    White
## 5:          0  35   Male 1/2/2011 Hispanic
```

```
dim(CCSO_small)
```

```
## [1] 5916    5
```

In this analysis we will investigate the propensity of incarcerations lasting longer than one day for crimes encoded as "other traffic offenses" where:

- The response variable is `atleastone` where a 1 indicates an incarceration lasting longer than one day and a 0 indicates an incarceration lasting shorter than 1 day.
- The covariates are: Age (age at arrest), Sex (Male or Female), Race (Asian/Pacific Islander, Black, Hispanic, and White).

Note: this data set is observational. We did have any control over who entered the data set or how long incarcerations were or anything else. **Why is this important to note?**

We can fit a basic main effects model in an instant using the glm function in R.

```
m1 <- glm(atleastone ~ -1 + Race + Sex + Age, data = CCSO_small,
          family = "binomial", x = "TRUE")
```

Now let's unpack the glm function call above. We decided that we wanted to fit an exponential family regression model with log likelihood taking the general form

$$l(\beta) = \langle M'y, \beta \rangle - c(M\beta),$$

where M specified by the formula in the glm function call above.

```
M <- m1$x
head(M)
```

```
##   RaceAsian/Pacific Islander RaceBlack RaceHispanic RaceWhite SexMale Age
## 1                          0         0            0         1       1  22
## 2                          0         0            0         1       1  26
## 3                          0         0            0         1       0  32
## 4                          0         0            0         1       1  22
## 5                          0         0            1         0       1  35
## 6                          0         0            1         0       1  35
```

The specific log likelihood for the logistic regression model can then be written as

$$l(\beta) \propto \sum_{i=1}^{n} y_i x_i^T \beta - \log\left(1 + \exp(x_i^T \beta)\right),$$

The glm function then performs a Fisher scoring based optimization routine (technically IRLS) to maximize the above likelihood. It stores $\hat{\beta}$ among many other useful quantities.

```
names(m1)
```

```
##  [1] "coefficients"      "residuals"         "fitted.values"
##  [4] "effects"           "R"                 "rank"
##  [7] "qr"                "family"            "linear.predictors"
## [10] "deviance"          "aic"               "null.deviance"
## [13] "iter"              "weights"           "prior.weights"
## [16] "df.residual"       "df.null"           "y"
## [19] "converged"         "boundary"          "model"
## [22] "x"                 "call"              "formula"
## [25] "terms"             "data"              "offset"
## [28] "control"           "method"            "contrasts"
## [31] "xlevels"
```

We can view summary information for $\hat{\beta}$ and the fitting process using the summary function

```
summary(m1)
```

```
##
## Call:
## glm(formula = atleastone ~ -1 + Race + Sex + Age, family = "binomial",
##     data = CCSO_small, x = "TRUE")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9393  -0.5485  -0.4817  -0.3391   2.6527
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## RaceAsian/Pacific Islander -4.389865   0.523612  -8.384  < 2e-16 ***
## RaceBlack                  -1.876550   0.144601 -12.977  < 2e-16 ***
## RaceHispanic               -2.804549   0.173349 -16.179  < 2e-16 ***
## RaceWhite                  -3.043226   0.147160 -20.680  < 2e-16 ***
## SexMale                     0.739834   0.105380   7.021 2.21e-12 ***
## Age                         0.007705   0.003186   2.418   0.0156 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 8201.3  on 5916  degrees of freedom
## Residual deviance: 4668.7  on 5910  degrees of freedom
## AIC: 4680.7
##
## Number of Fisher Scoring iterations: 6
```

Recall from the asymptotic theory of maximum likelihood estimation that

$$\sqrt{n}(\hat{\beta} - \beta) \xrightarrow{d} N(0, \Sigma^{-1}),$$

where Σ^{-1} is the inverse of the Fisher information matrix. We can extract these same standard errors using the vcov function

```
sqrt(diag(vcov(m1)))
```

```
## RaceAsian/Pacific Islander                 RaceBlack
##                 0.523611637               0.144600942
##               RaceHispanic                 RaceWhite
##                 0.173349091               0.147160126
##                    SexMale                       Age
##                 0.105379771               0.003186262
```

These values are the same as those in the Std. Error column in the above summary table

```
all.equal(summary(m1)$coef[, 2], sqrt(diag(vcov(m1))))
```

```
## [1] TRUE
```

Inference

Recall that we can make inferences about β_j using the Wald statistic corresponding to the hypothesis test

$$H_o : \beta_j = 0, \qquad H_a : \beta_j \neq 0,$$

which is given by

$$\frac{\hat{\beta}_j}{\text{se}(\hat{\beta}_j)} \sim N(0, 1),$$

We can compute the p-values by hand

```
Z <- abs(coef(m1)/sqrt(diag(vcov(m1))))
round(2*pnorm(Z, lower = FALSE), 4)
```

```
## RaceAsian/Pacific Islander                RaceBlack
##                     0.0000                   0.0000
##               RaceHispanic                RaceWhite
##                     0.0000                   0.0000
##                    SexMale                      Age
##                     0.0000                   0.0156
```

Deviance and likelihood ratio testing

Recall our $l(\mu; y)$ notation.

The deviance is defined by

$$-2\left[l(\hat{\mu}; y) - l(y; y)\right].$$

This is the likelihood-ratio for testing the null hypothesis that the model against the general alternative (ie, the saturated model). The deviance has reference distribution

$$-2\left[l(\hat{\mu}; y) - l(y; y)\right] \approx \chi^2_{\text{df}}$$

where $\text{df} = n - p$, n is the sample size, and p is the number of model parameters.

We do in fact obtain sufficient dimension reduction.

```
## compare with saturated model
m1$deviance
```

```
## [1] 4668.728
```

```
m1$df.residual
```

```
## [1] 5910
```

```
pchisq(m1$deviance, df = m1$df.residual, lower = FALSE)
```

```
## [1] 1
```

Can compute test by hand. First,

$$l(y; y) = \sum_{i=1}^{n} y_i \log(y_i) + (1 - y_i) \log(1 - y_i) = 0.$$

And $l(y; \hat{\mu})$ equals

```
logLik(m1)
```

```
## 'log Lik.' -2334.364 (df=6)
```

The deviance is the same

```
n <- nrow(CCSO_small); p <- length(coef(m1))
-2 * logLik(m1) == m1$deviance
```

```
## [1] TRUE
```

```
n - p == m1$df.residual
```

```
## [1] TRUE
```

```
as.numeric(pchisq(-2 * logLik(m1), df = n - p, lower = FALSE))
```

```
## [1] 1
```

Compare with null model

Fit the null model and perform test in R.

```
## use LRT testing in the anova function
m_null <- glm(atleastone ~ 1, family = "binomial", data = CCSO_small)
anova(m_null, m1, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: atleastone ~ 1
## Model 2: atleastone ~ -1 + Race + Sex + Age
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      5915     4982.7
## 2      5910     4668.7  5   313.99 < 2.2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Fit the model with $\hat{\mu} = \bar{y}$ and compute test by hand.

```
y <- CCSO_small$atleastone
prob <- mean(y)
round(-2 * sum(y*log(prob) + (1-y)*log(1-prob)), 3) == round(m_null$deviance, 3)
```

```
## [1] TRUE
```

```
Xstat <- -2 * sum(y*log(prob) + (1-y)*log(1-prob)) + 2 * logLik(m1)
pchisq(Xstat, df = p - 1, lower = FALSE)
```

```
## 'log Lik.' 9.807878e-66 (df=6)
```

A point about model matrices

Recall that span(M) is important in the sense that

$$\langle y, M_1\beta_1\rangle - c(M_1\beta_1) = \langle y, M_2\beta_2\rangle - c(M_2\beta_2)$$

when span(M_1) = span(M_2).

```
m1_2 <- glm(atleastone ~ Race + Sex + Age, data = CCSO_small,
          family = "binomial", x = "TRUE")
round(logLik(m1), 4) == round(logLik(m1_2), 4)
```

```
## [1] TRUE
```

```
coef(m1)
```

```
## RaceAsian/Pacific Islander                   RaceBlack
##                 -4.38986549                -1.87654964
##                RaceHispanic                   RaceWhite
##                 -2.80454861                -3.04322627
##                     SexMale                         Age
##                  0.73983403                 0.00770504
```

```
coef(m1_2)
```

```
##  (Intercept)    RaceBlack RaceHispanic    RaceWhite      SexMale          Age
##  -4.38986549   2.51331585   1.58531688   1.34663922   0.73983403   0.00770504
```

Now let's consider an interesting competing model

```
CCSO_small <- CCSO_small %>% mutate(isBlack = ifelse(Race == "Black",1,0))
m2 <- glm(atleastone ~ isBlack + Sex + Age, data = CCSO_small,
          family = "binomial", x = "TRUE")
summary(m2)
```

```
##
## Call:
## glm(formula = atleastone ~ isBlack + Sex + Age, family = "binomial",
##     data = CCSO_small, x = "TRUE")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9393  -0.5347  -0.4844  -0.3448   2.4245
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -3.030202   0.143625 -21.098  < 2e-16 ***
## isBlack      1.145529   0.075239  15.225  < 2e-16 ***
## SexMale      0.751374   0.104753   7.173 7.35e-13 ***
## Age          0.007656   0.003159   2.424   0.0154 *
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 4982.7  on 5915  degrees of freedom
## Residual deviance: 4684.1  on 5912  degrees of freedom
## AIC: 4692.1
##
## Number of Fisher Scoring iterations: 5
```

Is this new model nested within our old model? In other words, will

```
anova(m2, m1, test == "LRT")
```

work?

```
head(m2$x, 3)
```

```
##   (Intercept) isBlack SexMale Age
## 1           1       0       1  22
## 2           1       0       1  26
## 3           1       0       0  32
```

```
head(m1$x, 3)
```

```
##   RaceAsian/Pacific Islander RaceBlack RaceHispanic RaceWhite SexMale Age
## 1                          0         0            0         1       1  22
## 2                          0         0            0         1       1  26
## 3                          0         0            0         1       0  32
```

```
M <- cbind(rowSums(m1$x[, c(1,3,4)]), m1$x[, -c(1,3,4)])
head(M, 3)
```

```
##     RaceBlack SexMale Age
## 1 1         0       1  22
## 2 1         0       1  26
## 3 1         0       0  32
```

```
logLik(glm(y ~ M, family = "binomial"))
```

```
## 'log Lik.' -2342.04 (df=4)
```

```
logLik(m2)
```

```
## 'log Lik.' -2342.04 (df=4)
```

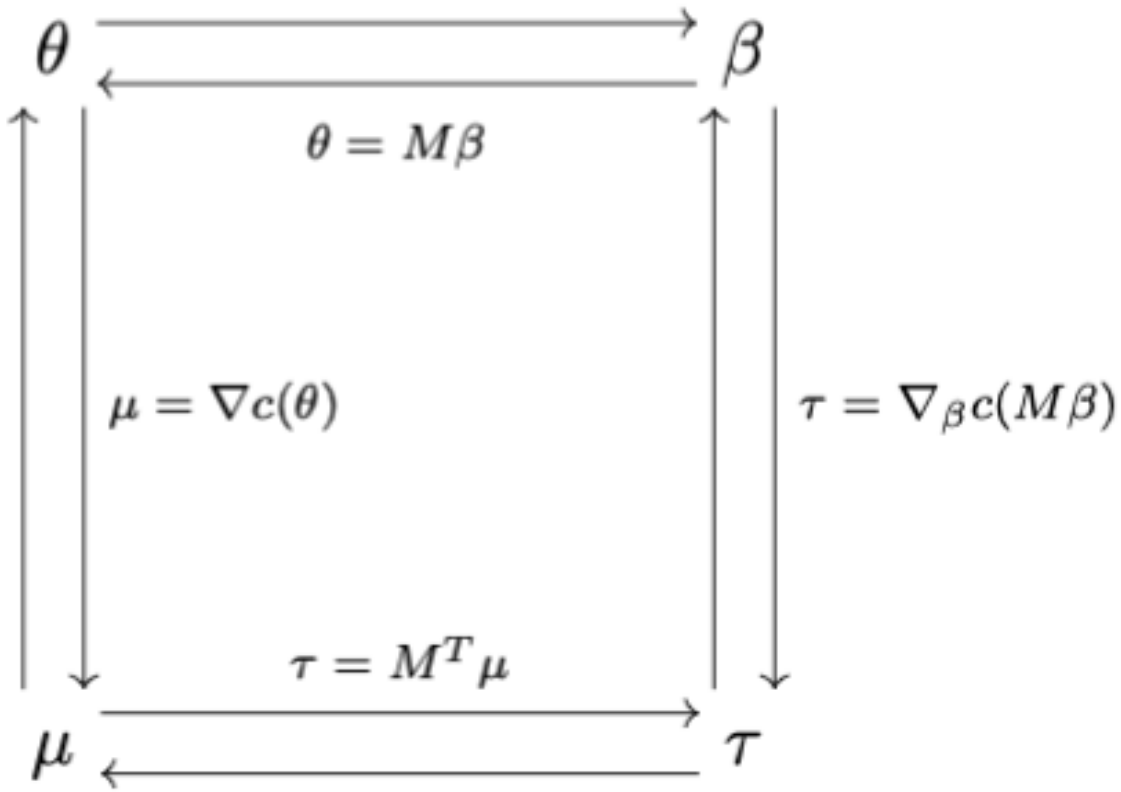
```r
anova(m2, m1, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: atleastone ~ isBlack + Sex + Age
## Model 2: atleastone ~ -1 + Race + Sex + Age
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      5912     4684.1
## 2      5910     4668.7  2   15.351 0.0004641 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' '
```

What does this test tell us?

Other parameterizations

Recall:



We start with

$$\langle M'y, \beta \rangle - c_\beta(\beta),$$

and then obtain $\hat{\beta}$ by maximizing the above. From here:

- $\hat{\theta} = M\hat{\beta}$
- $\hat{\mu} = \nabla_\theta c(\theta^*)|_{\theta* = \hat{\theta}}$
- $\hat{\tau} = M'\hat{u}$

For example we can compute $\hat{\theta}$ using the predict function or by hand.

```
theta <- m1$x %*% coef(m1)
head(cbind(predict(m1, type = "link"), theta), 3)
```

```
##        [,1]      [,2]
## 1 -2.133881 -2.133881
## 2 -2.103061 -2.103061
## 3 -2.796665 -2.796665
```

The submodel canonical parameterization scale is bit awkward for interpretation, although summary tables provide some insight to which components of the submodel canonical parameter vector may be driving the data generating process (under the assumed model).

R software provides functionality for estimating the mean value parameters

$$\mathsf{E}(Y|X = x) = \mathbb{P}(Y = 1|X = x),$$

associated with every individual in the study.

```
p1 <- predict(m1, type = "response", se.fit = TRUE)

## compute by hand
mu <- 1/(1 + exp(-theta))
head(cbind(p1$fit, mu), 3)
```

```
##         [,1]       [,2]
## 1 0.10584708 0.10584708
## 2 0.10879965 0.10879965
## 3 0.05750466 0.05750466
```

Another point about model matrices

Recall that model `m1_2` did not suppress the intercept.

This model's model matrix has the same span as the model that we have been working with. Mean-value parameters are the same for both models.

```
p1_2 <- predict(m1_2, type = "response", se.fit = FALSE)
head(cbind(p1$fit, p1_2), 6)
```

```
##                      p1_2
## 1 0.10584708 0.10584708
## 2 0.10879965 0.10879965
## 3 0.05750466 0.05750466
## 4 0.10584708 0.10584708
## 5 0.14245614 0.14245614
## 6 0.14245614 0.14245614
```