Binary Response Regression Notes

Daniel J. Eck

In these notes we discuss generalized linear models (GLMs) that arise natuarally from exponential family form. We will revisit the exponential family motivation and then present the logistic regression model for binary response data. We will also consider probit regression and propensity score estimation in causal inference.

Exponential family motivation

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 and x_i is a vector of predictors taking values in $\mathbb{R}^p$. 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 is a vector statistic having components y_i and $\theta \in \mathbb{R}^n$ 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.

We now motivate generalized linear models (GLMs), beginning with their formulation within the context of exponential family theory. These models were referred to as canonical linear submodels in the exponential family notes. The various parameterizations discussed in those notes arose naturally from general exponential family theory, and those parameterizations were linked to each other via explicit (or implicit) mappings which preserved maximum likelihood estimation. A canonical linear submodel of an exponential family is a submodel having parameterization

$$\theta = M\beta$$

where $\theta \in \mathbb{R}^n$ is the canonical parameter vector corresponding to the original saturated exponential family, $\beta \in \mathbb{R}^p$ is the canonical parameter vector for the submodel, M is the model matrix with rows x_i^T. The matrix M is usually called the *model matrix* in the terminology used by the `R` functions `lm` and `glm`. We will use this notation moving forward in these notes unless otherwise stated. The submodel log likelihood is

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

Recall that we have four parameters with relationships between them presented in Figure 1: the saturated model canonical and mean value parameters θ and μ and the canonical affine submodel canonical and mean value parameters β and $\tau = M^T\mu$. The observed equals expected property for the submodel is

$$\hat{\tau} = M^T\hat{\mu} = M^T y. \tag{3}$$

We cannot actually solve these equations for $\hat{\mu}$ because M the mapping $\mu \to M^T\mu$ is usually not one-to-one (the $n > p$ case where $M \in \mathbb{R}^{n\times p}$ and is full rank). Hence we cannot determine $\hat{\theta}$ and $\hat{\beta}$ from them either. The only way to determine the MLE is to maximize the log likelihood (2) for β to obtain $\hat{\beta}$ and then $\hat{\theta} = M\hat{\beta}$ and $\hat{\mu} = \nabla c(\hat{\theta})$ and $\hat{\tau} = M^T\hat{\mu}$.

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 reparameterize $\theta = M\beta$ and write

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

which implies that we can write

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

Therefore, a linear function of the canonical submodel parameter vector is linked to the mean of the exponential family through the inverse change-of-parameter mapping g^{-1}. This is the basis of exponential family generalized linear models with link function g^{-1}.

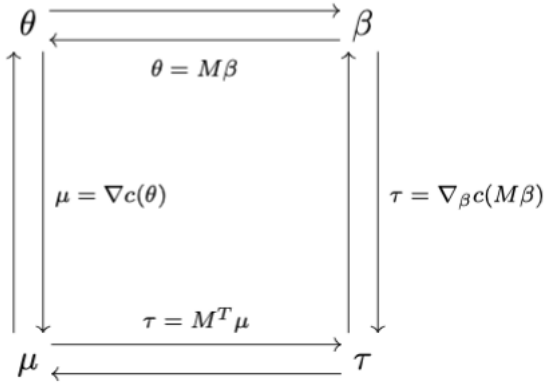


Figure 1: A depiction of the transformations necessary to change between parameterizations. Arrows going in opposite directions specify transformations and their inverses. M is a known model matrix of full column rank, a is a known offset vector, and c is the cumulant function for the exponential family model.

Logistic regression model

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

$$\pi(x) = P(Y = 1|X = x) = 1 - P(Y = 0|X = x).$$

The logistic regression model is then

$$\pi(x) = P(Y = 1|X = x) = \frac{\exp(x^T\beta)}{1 + \exp(x^T\beta)} = \frac{1}{1 + \exp(-x^T\beta)}. \tag{4}$$

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

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

In vector notation, we can express the above as

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

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

To see where the logit link in the specification of $\pi(x)$ comes from, consider the log likelihood of the binomial distribution

$$\begin{aligned} l(\beta) &\propto \sum_{i=1}^{n} y_i \log\left(\frac{p_i}{1-p_i}\right) + \sum_{i=1}^{n} \log(1-p_i) \\ &= \sum_{i=1}^{n} y_i \theta_i + \sum_{i=1}^{n} \log(1-g(\theta_i)), \end{aligned}$$

where

$$\theta_i = \log\left(\frac{p_i}{1-p_i}\right) \qquad \text{and} \qquad p_i = g(\theta_i) = \frac{\exp(\theta_i)}{1+\exp(\theta_i)} = \frac{1}{1+\exp(-\theta_i)}.$$

This implies that $g = \pi$ and we see that the logistic regression model is the same as the canonical linear submodel of an exponential family with $\theta_i = x_i^T \beta$ which in vector notation is $\theta = M\beta$.

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).

Example: CCSO data

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

We will demonstrate logistic regression modeling on the Champaign County Sheriff's Office (CCSO) data frame. This data frame consists of $n = 67764$ observations and 35 variables in total. We will only consider a few of these variables, and only consider observations with complete cases.

We now load in the CCSO data frame using the fread (**f**ast **read**) function from the data.table package and perform most of our data wrangling operations using functionality in the tidyverse. Note these data wrangling steps can also be performed using functionality in the data.table package or using basic R. We use the tidyverse because it provides a framework that allows one to read and write code with relative ease. Some may say that the data.table provides a more powerful set of tools for handling large data sets, but these debates are beyond the scope of this lecture. You may want to play around with both!

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

```
##    user  system elapsed
##   0.275   0.085   4.901
```

```
## 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")) %>%
  ## Agency (and possible interactions with Agency) is interesting if you want to go deeper
  #filter(Agency %in% c("Champaign County Sherriff's Office", "Champaign Police Department",
  #                     "Illinois State Police", "Rantoul Police Department",
  #                     "University of Illinois Police Department", "Urbana Police Department")) %>%
  #dplyr::select(atleastone, Age, Sex, Date, Race, Agency) %>%
  #mutate(Race = fct_drop(Race), Sex = fct_drop(Sex), Agency = fct_drop(Agency))
  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)
```

```
##    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
## 6:          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". 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. We will also determine whether or not any demographic variables drive the propensity of incarcerations lasting longer than one day. The variables under consideration are: Age (age at arrest), Sex (Male or Female), Race (Asian/Pacific Islander, Black, Hispanic, and White).

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

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

where y is the vector of responses β is the submodel canonical statistic vector corresponding to the model matrix M specified by the formula in the glm function call above. The first few rows of M are displayed below

```
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

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

where the x_is are the rows of the design matrix M and the y_is are the components of the response vector y (the `atleastone` variable corresponding to incarcerations lasting longer than one day). 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, some of which will be discussed shortly. 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
```

The Estimate column in the above summary table is $\hat{\beta}$. The standard error column contains estimates of the square root of the variances of the estimated submodel canonical parameter vector $\hat{\beta}$. 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
```

Manually write your own Fisher scoring algorithm which maximizes the logistic regression log likelihood for this example. Report $\hat{\beta}$ and reproduce the above summary table without using the glm or summary commands.

Inference for β in logistic regression

preliminaries

The CCSO example connects the exponential family that we have developed to R-based data analysis. We will now discuss inference in logistic regression models. We begin with β. Recall that β exhibits a linear relationship when the response variable takes the form $\text{logit}(\pi(x))$, where $\pi(x)$ is the conditional probability of success at covariate value x.

The logit function has another names, the log-odds ratio. Therefore, a unit increase in one predictor variables x_j corresponds to an increase of β_j (estimated by $\hat{\beta}_j$) in the log-odds ratio with everything else being held fixed. This interpretation is the same as that in linear regression, except it is harder. A unit increase in x_j corresponds to an increase of β_j (estimated by $\hat{\beta}_j$) increase in the log of the ratio of the probability of success to the probability of failure.

HW Derive the log-odds ratio of $x+1$ to x when $Y=1$, and observe that the log-odds ratio does not depend on x.

The odds ratio itself is hard to interpret, so in many cases simple heuristics suffice. Just like in linear regression, the following rules hold:

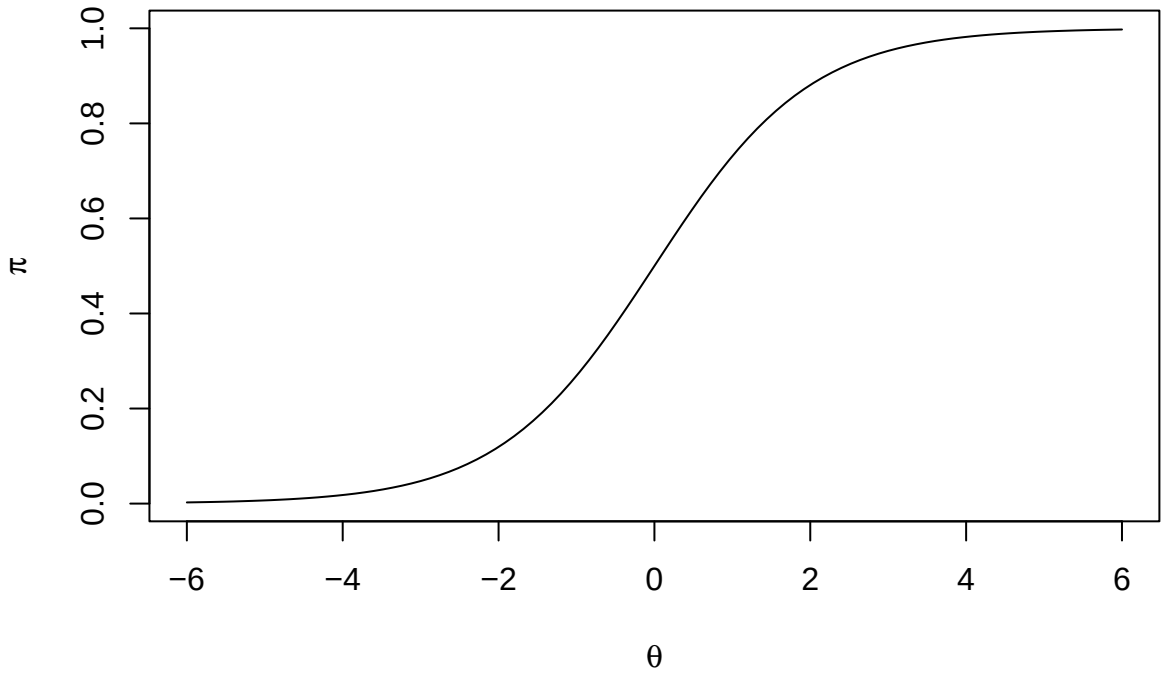
- $\beta > 0$: increasing X implies that $\mathbb{P}(Y=1|X=x)$ increases.
- $\beta = 0$: changing X implies that $\mathbb{P}(Y=1|X=x)$ do not change.
- $\beta < 0$: increasing X implies that $\mathbb{P}(Y=1|X=x)$ decreases.

Note that the similarities between logistic and linear regression models stop when we discuss the relationship between β and $\pi(x) = \mathbb{P}(Y=1|X=x)$. In linear regression there is no "second parameterization," the link function g^{-1} is the identity function, so the slope $\nabla_x E(Y|X=x) = \beta$ in linear regression. In logistic regression, we have that

$$\nabla_x \pi(x) = \nabla_x \mathrm{E}(Y|X=x) = \beta\pi(x)(1-\pi(x)),$$

where $\mathrm{E}(Y|X=x) = \mathbb{P}(Y=1|X=x)$. The relationship between $\pi(x)$ and the linear predictor $\theta(x) = x^T\beta$ is shown below

```
library(faraway)
curve(ilogit(x),-6,6, xlab=expression(theta), ylab=expression(pi))
```



The logistic curve is almost linear in its midrange. This means that for modeling responses that are all of moderate probability, logistic and linear regression will not behave very differently. The curve approaches one at its upper end and zero at its lower end but never reaches these bounds. This means that logistic regression will never predict that anything is inevitable or impossible.

Now that we have a decent idea of what the submodel canonical parameter vector β is in a logistic regression model we can discuss statistical inference corresponding to estimates of β.

Wald inference for regression coefficients

As in linear regression 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

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

where this distributional relationship holds under the null hypothesis $\beta_j = 0$. Similarly, we can form a confidence interval

$$\hat{\beta}_j \pm z_{\alpha/2}\mathrm{se}(\hat{\beta}_j)$$

where $0 < \alpha < 1$ is some error threshold. In a similar vein, we can construct $(1-\alpha)\times 100\%$ confidence intervals for the odds ratio (not log-odds ratio) e^{β_j} given as

$$(e^{L_j}, e^{U_j})$$

where $L_j = \hat{\beta}_j - z_{\alpha/2}\mathrm{se}(\hat{\beta}_j)$ and $U_j = \hat{\beta}_j + z_{\alpha/2}\mathrm{se}(\hat{\beta}_j)$ when the Wald confidence interval is reported.

The $(1-\alpha)\times 100\%$ Wald confidence interval for the response variable $\text{logit}(\pi(x))$ at a particular x follows from the Delta method. We have that $\text{logit}(\pi(x)) = x^T\beta$ which implies that $\nabla_\beta \text{logit}(\pi(x)) = x$. Therefore,

$$\sqrt{n}\left(\text{logit}(\hat{\pi}(x)) - \text{logit}(\pi(x))\right) \xrightarrow{d} N(0, x^T\Sigma^{-1}x).$$

We can report the following $(1-\alpha)\times 100\%$ Wald confidence interval for the log-odds ratio

$$\text{logit}(\hat{\pi}(x)) \pm z_{\alpha/2}\sqrt{x^T\widehat{\Sigma}^{-1}x}.$$

Note that, while commonly used in practice, Wald inference can be suboptimal in certain settings. See Section 5.2.6 of Agresti [2013] for specifics.

Deviance and likelihood ratio testing

Let $l(\mu; y)$ denote the the log-likelihood of a GLM in the mean-value parameter μ. Let $l(\hat{\mu}; y)$ denote the MLE of the log likelihood for the model. The saturated model with $\hat{\mu} = y$ has the maximum achievable log likelihood $l(y; y)$. This model is not useful in practice, but it does allow for comparison to other model fits. 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. For large samples we can use the deviance statistic to test nested models. Denote $\mathcal{M}_o$ and $\mathcal{M}_a$, respectively, as the null model and the alternative model, and let $\hat{\mu}_o$ and $\hat{\mu}_a$, respectively, be the estimated mean-value parameter vectors under $\mathcal{M}_o$ and $\mathcal{M}_a$. Then the difference in deviances between the null model and the alternative model is,

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

where $\text{df} = p_a - p_o$, p_a is the number of model parameters in model $\mathcal{M}_a$, and p_o is the number of model parameters in model $\mathcal{M}_o$.

Example: CCSO data (continued)

We are now equipped to understand the summary table that is returned by summary.glm.

```
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
```

We can use the deviance and degrees of freedom objects to perform the likelihood ratio test to determine whether or not the main effects model fits the data well. In this case where the intercept is suppressed, the null model forces the intercept to be zero. This is equivalent to assuming that the Bernoulli success probability is fixed at $p = 1/2$ for every subject.

```
## compare with saturated model
pchisq(m1$deviance, df = m1$df.residual, lower = FALSE)
```

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

```
## use Chi squared testing to directly compare submodels
# null deviance
m1$null.deviance
```

```
## [1] 8201.317
```

```
# deviance of model with p = 1/2
-2 * 5916 * log(1/2)
```

```
## [1] 8201.317
```

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

```
## [1] 0
```

```
## use LRT testing in the anova function
# here the null model allows for an intercept term to be present
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
```

Let's consider the smaller model that ignores the Sex variable. A likelihood ratio test shows that the larger model is preferable at any reasonably chosen significance level α.

```r
m_small <- glm(atleastone ~ -1 + Race + Age, family = "binomial", data = CCSO_small)

## built in likelihood ratio test using anova.glm
anova(m_small, m1, test = "LRT")
```

```
## Analysis of Deviance Table
##
## Model 1: atleastone ~ -1 + Race + Age
## Model 2: atleastone ~ -1 + Race + Sex + Age
##   Resid. Df Resid. Dev Df Deviance  Pr(>Chi)
## 1      5911     4724.4
## 2      5910     4668.7  1   55.709 8.401e-14 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

```r
## perform the above directly
pchisq(55.709, df = 1, lower = FALSE)
```

```
## [1] 8.403269e-14
```

Summarize the information about β in the above summary table produced by a call to summary(m1). Keep in mind that we restricted attention to crimes classified as "other traffic offenses".

Mean-value parameters

The 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 $\pi(x) = \mathrm{E}(Y|X = x) = \mathbb{P}(Y = 1|X = x)$, associated with every individual in the study. The function used to obtain mean value parameter estimates is `predict.glm` with `type = "response"` specified

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

Under the hood, we specified a model with a formula call and a data frame. A model matrix M is then created and $\hat{\beta}$ is estimated with the initial call to `glm`. The saturated model canonical parameter is expressed as $\hat{\theta} = M\hat{\beta}$, and this results in estimated mean value parameters of the form $\hat{\mu} = \nabla c(\hat{\theta})$. Recall that the submodel log likelihood can be expressed as

$$\begin{aligned} l(\beta) &\propto \sum_{i=1}^{n} y_i x_i^T \beta - \log\left(1 + \exp(x_i^T \beta)\right) \\ &= \sum_{i=1}^{n} y_i \theta_i - \log\left(1 + \exp(\theta_i)\right) = l(\theta) \end{aligned}$$

where $\theta_i = x_i^T \beta$. Therefore the mean value parameters are

$$\nabla c(\theta) = \nabla \log\left(1 + \exp(\theta)\right) = \frac{e^{\theta}}{1 + e^{\theta}},$$

which are estimated by plugging in $\hat{\theta} = M\hat{\beta}$ to the above. This is what `predict.glm` does to obtain $\hat{\mu}$ for a model with model matrix M.

```
## estimates from GLM
p1.fit <- as.numeric(p1$fit)

## estimates from hand
betahat <- m1$coefficients
foo <- as.numeric(exp(M %*% betahat) / (1 + exp(M %*% betahat)))

## the same
head(cbind(p1.fit, foo))
```

```
##          p1.fit        foo
## [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
```

```
all.equal(p1.fit, foo)
```

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

Good, they are in fact equal.

Standard errors $\text{se}(\hat{\mu})$ that are computed from `predict.glm` also follow from straighforward exponential family results. **This is left as a homework assignment. Hint: check your results against what is produced by a call to predict.glm with the argument se.fit = TRUE.**

There are noted problems with Wald based confidence intervals for mean value parameters in logistic regression (the success probability). See here for more details in the context of binomial sampling proportions.

Profile likelihood based confidence intervals

Say that we want a confidence interval for a parameter vector β in a model $\mathcal{M}$. Let $\mathcal{M}_o(\beta_o)$ be the same model as $\mathcal{M}$, except that β is fixed at β_o. Then the likelihood ratio tests the hypothesis that

$$H_o : \beta = \beta_o \qquad H_a : \beta \neq \beta_o$$

and this test produces a P-value (from a χ^2 approximation). Then

$$\{\beta_o : \text{P-value} > \alpha\}$$

is a $(1 - \alpha) \times 100\%$ confidence set (usually a confidence interval) for β. This interval is a profile likelihood confidence interval. Standard software estimates this interval.

```
## profile based method
beta.profile <- confint(m1)
```

```
## Waiting for profiling to be done...
```

```
beta.profile
```

```
##                                  2.5 %      97.5 %
## RaceAsian/Pacific Islander -5.59295728 -3.48648322
## RaceBlack                  -2.16230780 -1.59523096
## RaceHispanic               -3.14837654 -2.46857072
```

```
## RaceWhite                  -3.33481810 -2.75773448
## SexMale                     0.53693218  0.95034741
## Age                         0.00141719  0.01391126
```

```
## Wald based method
se <- sqrt(diag(vcov(m1)))
cbind(betahat + qnorm(0.025) * sqrt(diag(vcov(m1))),
      betahat + qnorm(0.975) * sqrt(diag(vcov(m1))))
```

```
##                                   [,1]       [,2]
## RaceAsian/Pacific Islander -5.416125438 -3.3636055
## RaceBlack                  -2.159962276 -1.5931370
## RaceHispanic               -3.144306583 -2.4647906
## RaceWhite                  -3.331654813 -2.7547977
## SexMale                     0.533293472  0.9463746
## Age                         0.001460082  0.0139500
```

Diagnostics

Model diagnostics for logistic regression fits are not as prevalent as those in linear regression.

Esarey and Pierce [2012] developed a method to assess the fit of binary-dependent variable models and compare this method to existing approaches. This method compares a model's predicted probability $\mathbb{P}(Y = 1)$ to the observed frequency of $Y = 1$ in the data set. If the model is a good fit to the data, subsets of the data with $\mathbb{P}(Y = 1) \approx m$ should have about m proportion of cases for which $Y = 1$. The process is analogous to plotting fitted and observed values of the dependent variable against one another, a commonly used fit diagnostic for continuous dependent variable models.

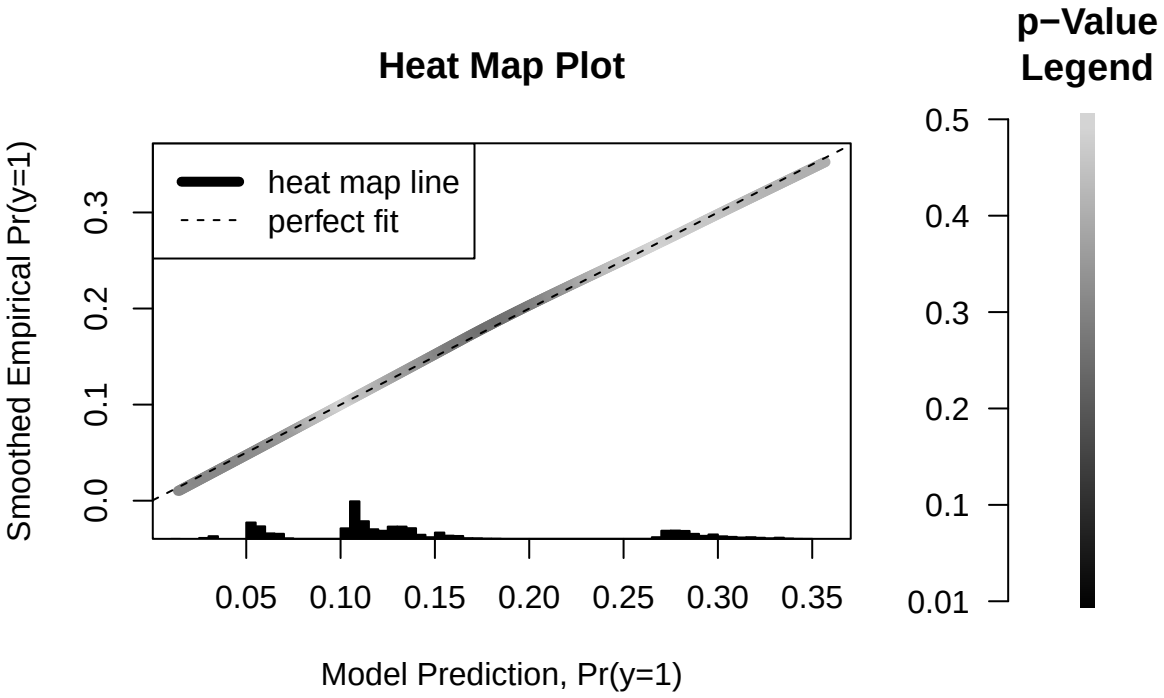
The difficulty is that (unlike Y) the true success probability p is not observable. But if we collect all the observations with values of $\hat{p} \approx m$, approximately m proportion of those cases should be $Y = 1$ when $\hat{p}$ is a good estimate of p. That is, $\mathbb{P}(Y = 1|\hat{p} = m) \approx m$ if our model is a good predictor of probability.

In brief, the approach in Esarey and Pierce [2012] is to estimate an empirical model, sort observations according to the predicted p of the model, and then determine whether this predicted probability is an accurate predictor of $R(p) = \mathbb{P}(Y = 1|p)$ by plotting them against each other. In a perfectly fitting model, $p = R(p)$ for all values of p and the plot is a straight 45 degree line through the origin.

This methodology is implemented in the `heatmapFit` package. We see that our fitted model exhibits great fit.

```
library(heatmapFit)
y <- CCSO_small$atleastone
heatmap.fit(y, p1.fit)
```

```
##
## Calculating optimal loess bandwith...
## aicc Chosen Span =  0.9899469
##
## Generating Bootstrap Predictions...
##   |                                                                      |
```

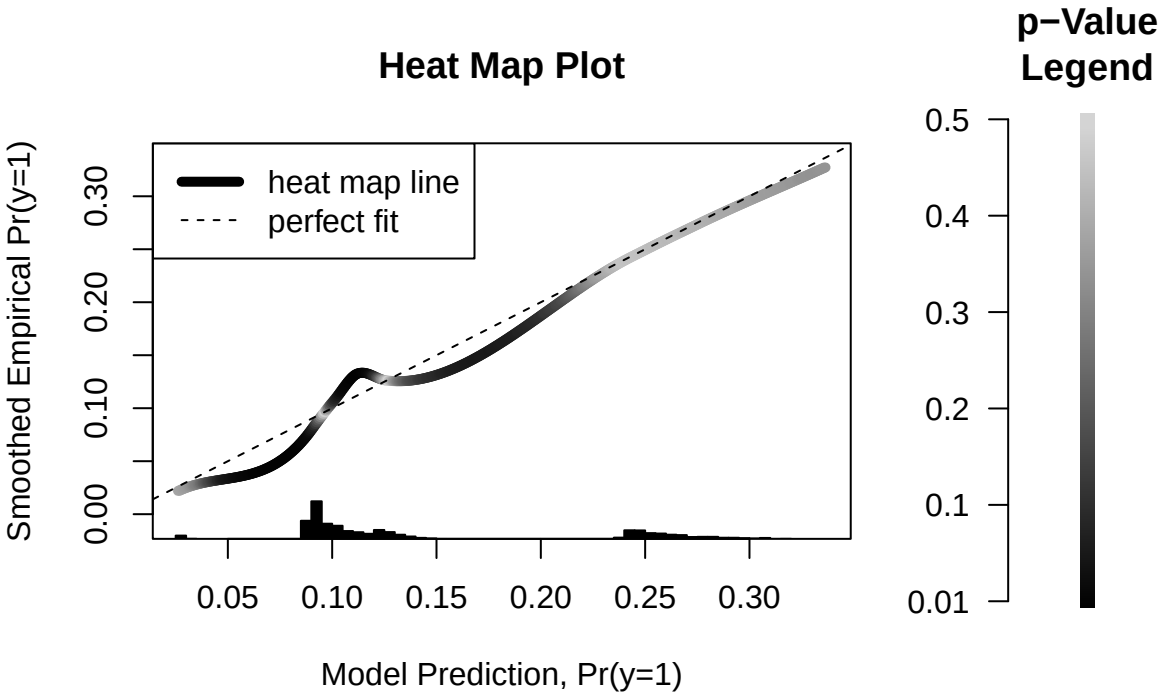



```
##
##
## ****************************************
## 0% of Observations have one-tailed p-value <= 0.1
## Expected Maximum = 20%
## ****************************************
```

We can see that the model which omits Sex does not fit the data as well.

```
heatmap.fit(y, predict(m_small, type = "response"))
```

```
##
## Calculating optimal loess bandwith...
## aicc Chosen Span =  0.6419477
##
## Generating Bootstrap Predictions...
##   |                                                                      |
```




```
##
##
## ****************************************
## 32.25152% of Observations have one-tailed p-value <= 0.1
## Expected Maximum = 20%
## ****************************************
```

Alternatives to the logit link

Probit Regression

In this class we have focused on exponential theory, canonical links, and generalized linear models that arise from this perspective. We now present an alternative to this paradigm in the form of the probit link function for binary response variables. Probit regression arises from normal latent variable models. Recall that we are interested in modeling $\pi(x) = \mathbb{P}(Y = 1|X = x)$ in the form

$$g^{-1}(\pi(x)) = x^T\beta$$

where g^{-1} is not the logit function. In probit regression we specify $g = \Phi$ where Φ is the CDF of a standard normal distribution. Thus the probit regression model is of the form

$$\Phi^{-1}(\pi(x)) = x^T\beta.$$

A related latent variable (referred to as a threshold model) also specifies the probit model. This model assumes that there is an unobserved continuous response y^* such that the observed response $y = 0$ if $y^* \leq \tau$ and $y = 1$ if $y^* > \tau$. Suppose that $y^* = \mu + \varepsilon$, where $\mu = x^T\beta$ and where ε_i are independent realizations from a $N(0, \sigma^2)$ distribution. Then,

$$\begin{aligned}\mathbb{P}(Y = 1 \mid X = x) &= \mathbb{P}(Y^* > \tau \mid X = x) = \mathbb{P}(x^T\beta + \varepsilon > \tau \mid X = x) \\ &= \mathbb{P}(-\varepsilon < x^T\beta - \tau \mid X = x) = \Phi\left((x^T\beta - \tau)/\sigma\right)\end{aligned}$$

where it is noted that $\varepsilon \stackrel{d}{=} -\varepsilon$. There is no information in the data about σ or the threshold τ. An equivalent model results if we multiply (β^T, σ, τ) by any positive constant. For identifiability, we set $\sigma = 1$ and $\tau = 0$, and the probit model results. It should be noted that the same latent variable threshold model story can be told by the logistic regression model. To see this, replace the normal distribution specification for ε in the above with with a logistic distribution with $\mu = 0$ and $s = 1$.

Another latent variable interpretation of the probit model is based on utility functions. Consider the choice between two options, such as two brands. Let U_0 denote the utility of outcome $y = 0$ and let U_1 denote the utility of outcome $y = 1$. For $y \in \{0, 1\}$, suppose that $U_y = x^T\beta_y + \varepsilon_y$. A particular subject selects $y = 1$ when $U_1 > U_0$. Now suppose that ε_0 and ε_1 are independent $N(0, 1)$ random variables. Then,

$$\begin{aligned}\mathbb{P}(Y = 1 \mid X = x) &= \mathbb{P}(x^T\beta_1 + \varepsilon_1 > x^T\beta_0 + \varepsilon_0 \mid X = x) \\ &= \mathbb{P}\left((\varepsilon_0 - \varepsilon_1)/\sqrt{2} < x^T(\beta_1 - \beta_0)/\sqrt{2} \mid X = x\right) \\ &= \Phi(x^T\beta^*)\end{aligned}$$

where $\beta^* = (\beta_1 - \beta_0)/\sqrt{2}$. Again, this is the probit model. The probit model can extend to an inverse t link, for which corresponding latent variable models can better accommodate outliers.

Parameters in probit models can be interpreted in terms of effects on $\mathrm{E}(Y^*)$ for the threshold latent variable model presented above. Since $Y^* = x^T\beta + \varepsilon$ where $\varepsilon \sim N(0, 1)$ has the CDF Φ, a 1-unit increase in x corresonds to a β standard deviation increase in $\mathrm{E}(Y^*)$. Alternatively, we can summarize effects on the probability scale, such as by comparing estimated probabilities at extreme values or quartiles of a predictor, with other predictors set at their means. Although probit model parameter estimates are on a different scale than logistic model parameter estimates, the probability summaries of effects are similar.

Probit model fitting

Let y_i be the number of successes out of n_i trials at setting x_i, $i = 1, \ldots, n$. Let x_{ij} denote the value of predictor j for subject i. For the probit model $\Phi^{-1}(\pi(x_i)) = \sum_{j=1} \beta_j x_{ij}$, the log likelihood function is

$$l(\beta) = \log\left\{\prod_{i=1}^{n}\left[\Phi\left(\sum_j \beta_j x_{ij}\right)\right]^{y_i}\left[1 - \Phi\left(\sum_j \beta_j x_{ij}\right)\right]^{n_i - y_i}\right\}.$$

Differentiation with respect to β_j leads to,

$$\sum_i \left\{\left[\frac{n_i\left[y_i - \Phi\left(\sum_j \beta_j x_{ij}\right)\right]x_{ij}}{\Phi\left(\sum_j \beta_j x_{ij}\right)\left[1 - \Phi\left(\sum_j \beta_j x_{ij}\right)\right]}\right]\phi\left(\sum_j \beta_j x_{ij}\right)\right\} = 0,$$

with $\phi(\cdot)$ the standard normal pdf.

When the link function is not the canonical link (which is the logit function for binary data), there is no reduction of the data in the form of Fisher's notion of sufficieny. Fisher (1935), in an appendix to Bliss (1935) for the single predictor case, showed how to solve the above equations using the Fisher scoring algorithm. He also pointed out that cases with $y_i = 0$ or $y_i = n_i$ were not problematic for ML fitting, unlike weighted least squares using sample probits (or logits). The estimated asymptotic covariance matrix of $\hat{\beta}$ has the form

$$\widehat{\mathrm{cov}}(\hat{\beta}) = \left(M^T\widehat{W}M\right)^{-1}.$$

For probit models, $\widehat{W}$ is the diagonal matrix with elements

$$w_i = \frac{n_i\left[\phi\left(\sum_j \hat{\beta}_j x_{ij}\right)\right]^2}{\left\{\Phi\left(\sum_j \hat{\beta}_j x_{ij}\right)\left[1 - \Phi\left(\hat{\beta}_j x_{ij}\right)\right]\right\}}.$$

The Newton-Raphson algorithm yields the same MLE estimates but slightly different standard errors. For the information matrix inverted to obtain the asymptotic covariance matrix, Newton-Raphson uses observed information, whereas Fisher scoring uses expected information. These differ for link functions other than the canonical link.

Probit Example

We return to our CCSO example and refit the logistic regression model with a probit link function.

```
m2 <- glm(atleastone ~ -1 + Race + Sex + Age, data = CCSO_small,
          family = binomial(link = "probit"), x = "TRUE")
summary(m2)
```

```
##
## Call:
## glm(formula = atleastone ~ -1 + Race + Sex + Age, family = binomial(link = "probit"),
##     data = CCSO_small, x = "TRUE")
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -0.9309  -0.5530  -0.4830  -0.3303   2.6569
##
## Coefficients:
##                              Estimate Std. Error z value Pr(>|z|)
## RaceAsian/Pacific Islander -2.379296   0.228047 -10.433  < 2e-16 ***
## RaceBlack                  -1.093877   0.077919 -14.039  < 2e-16 ***
## RaceHispanic               -1.619120   0.092273 -17.547  < 2e-16 ***
## RaceWhite                  -1.741257   0.077547 -22.454  < 2e-16 ***
## SexMale                     0.394031   0.054464   7.235 4.66e-13 ***
## Age                         0.004492   0.001763   2.548   0.0108 *
## ---
## 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: 4668.3  on 5910  degrees of freedom
## AIC: 4680.3
##
## Number of Fisher Scoring iterations: 5
```

Summarise these results.

Inverse propensity score weighting (IPW) in Causal Inference

One of the central goals of causal inference is to estimate the average treatment effect (ATE). In its most simple presentation, we will assume that the treatment variable is binary. The ATE measures the difference in mean outcomes between individuals assigned to the treatment and individuals assigned to the control. However, there is a difficulty with estimation since individuals do not simultaneously receive the treatment and the control. When the study is randomized this difficulty is mitigated because randomization ensures (when n is large enough) that baseline covariates which may influence the response have the same distribution in both the treatment and control groups. In observational studies this may not be so, and the researcher may have no information about the treatment assignment mechanism. In the observational setting, the collected covariates could confound the relationship between the treatment variable and the response variable.

One technique for estimating the ATE in causal studies is through inverse propensity score weighting (IPW). The IPW technique makes use of a treatment assignment model. In the simple setting we will take a logistic regression model as our model for treatment assignment. The idea of IPW is to create a pseudo population that exhibits balance in baseline covariates through the weighting of individuals corresponding to how likely they belong to either the treatment or control group given their covariate profile. Thus IPW can mimic random assignment up to measured confounders.

We will now go through an example. In this example we estimate the causal effect of online learning in STAT 200 at UIUC. The response variable Y is the comprehensive 3-hour Final that was Scantron graded at the end of each semester, and the treatment variable A is the presence or absence of in-person lectures. The online only course is considered the treatment and the regular in-person course with recorded lectures is the control.

We estimate the causal effect of online learning (or absence of in-person lecture) by estimating the ATE using inverse propensity score weighting methods. The form of this estimator of the ATE is

$$\widehat{\text{ATE}} = \frac{1}{n}\sum_{i=1}^{n}\left(w_i A_i Y_i - w_i(1-A_i)Y_i\right),$$

where the weights w_i are a functions of the propensity scores $\hat{p}_i = \widehat{P}(A_i = 1|X_i)$, $i = 1, \ldots, n$. The propensity scores are each subjects conditional probability of belonging to the treatment group given their covariate information. We will estimate the propensity scores using a logistic regression model.

The most basic inverse propensity score weighted estimator has weights w_i of the form

$$w_i = \frac{A_i}{\hat{p}_i} + \frac{1-A_i}{1-\hat{p}_i}.$$

This estimator is unstable, it does not create balanced pseudo populations, and is inappropriate for continuous predictors in our analysis. We consider an alternative stable estimator of the ATE (refered to as ATE_{IPW}). The alternative stable estimator of the ATE has conventional inverse propensity score weights with changes being made to the aggregation. This estimator of the ATE is

$$\widehat{\text{ATE}}_{\text{alt}} = \left(\sum_{i=1}^{n}\frac{A_i}{\hat{p}_i}\right)^{-1}\sum_{i=1}^{n} w_i A_i Y_i - \left(\sum_{i=1}^{n}\frac{1-A_i}{1-\hat{p}_i}\right)^{-1}\sum_{i=1}^{n} w_i(1-A_i)Y_i$$

where w_i in the above are basic IPW weights.

The estimators $\widehat{\text{ATE}}$ and $\widehat{\text{ATE}}_{\text{alt}}$ are consistent if the propensity score model is correctly specified, so should be approximately unbiased in finite samples. We will also consider a double robust (DR) estimator of the ATE. This estimator is

$$\widehat{\text{ATE}}_{\text{DR}} = \frac{1}{n}\sum_{i=1}^{n}\frac{A_i Y_i - (A_i - \hat{p}_i)m_1(X_i, \hat{\alpha}_1)}{\hat{p}_i} - \frac{1}{n}\sum_{i=1}^{n}\frac{(1-A_i)Y_i + (A_i - \hat{p}_i)m_0(X_i, \hat{\alpha}_0)}{1-\hat{p}_i},$$

where $m_a(X, \alpha_a) = E(Y|A = a, X)$ is the regression of the response on X in each level of $A \in \{0, 1\}$, depending on parameters α_a, and $\hat{\alpha}_a$ is the estimator for α_a based on the data from subjects with $A = a$.

We now load in the data.

```
dat <- read.csv("online.csv", header = TRUE)[, -1]
head(dat)
```

```
##   ObjExam Online ACTMath ACTMajor  ACT Gender International F17 S18 S19 Fa19 FR
## 1   86.75      1    31.0     31.5 30.0      0             0   1   0   0    0  0
## 2   91.57      1    35.6     31.5 33.2      0             0   1   0   0    0  0
## 3   86.75      1    35.6     33.5 33.9      0             1   1   0   0    0  0
## 4   96.39      0    36.0     33.5 35.0      0             0   1   0   0    0  0
## 5   78.31      0    35.0     31.5 30.0      0             1   1   0   0    0  1
```

```
## 6   67.47      1    33.0     30.5 33.0      1             0   1   0   0    0  0
##   SO JR
## 1  0  1
## 2  1  0
## 3  0  0
## 4  1  0
## 5  0  0
## 6  0  1
```

We specify the propensity score logistic regression model and obtain the estimated propensity scores (estimates of the probability of being in the treatment group).

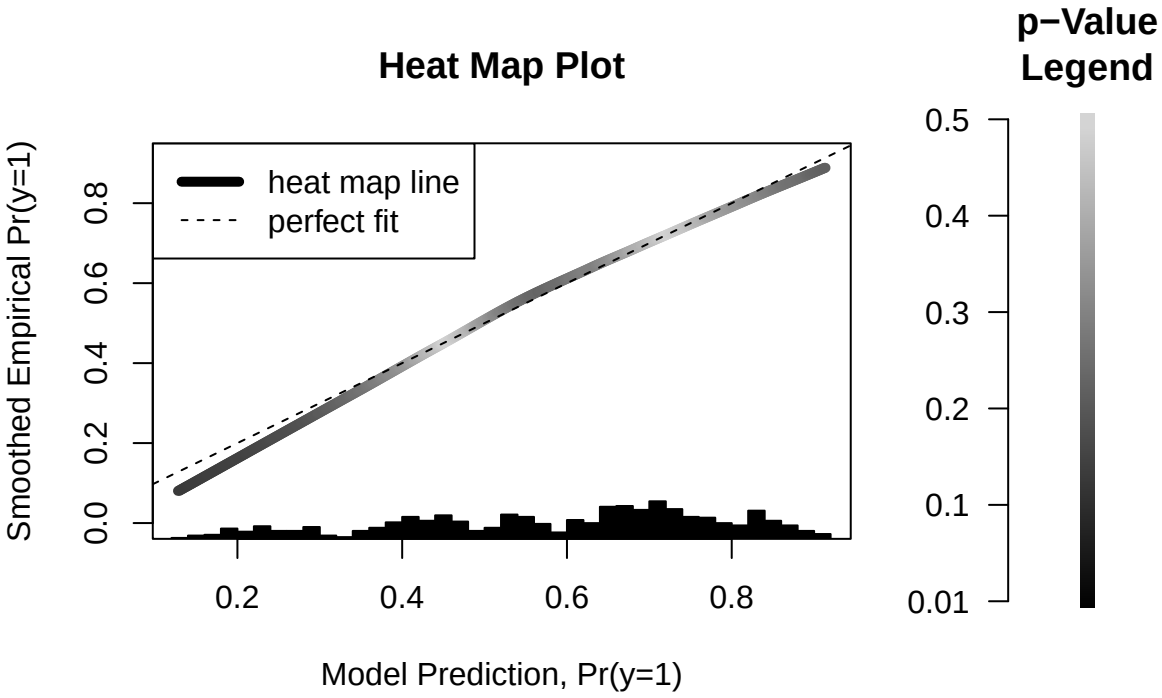
```
# basic data wrangling (will induce non problematic rank deficiency)
dat_small <- dat %>% dplyr::select(Online, ACTMath, ACTMajor, ACT, Gender,
                        International, F17, S18, S19, Fa19, FR, SO, JR)

# prop score model
m <- glm(Online ~., data = dat_small, family = "binomial")
trt <- dat_small$Online
preds <- predict(m, type = "response")
```

The propensity score model seems to align well with the observed data.

```
trt <- dat_small$Online
heatmap.fit(trt, preds)
```

```
##
## Calculating optimal loess bandwith...
## aicc Chosen Span =  0.9867739
##
## Generating Bootstrap Predictions...
##   |                                                                      |
```




```
##
##
## ****************************************
## 0% of Observations have one-tailed p-value <= 0.1
## Expected Maximum = 20%
## ****************************************
```

We now obtain the weights for the stabilized estimator

$$\widetilde{w}_i = \frac{\frac{A_i}{\hat{p}_i}}{\sum_{i=j}^{n} \frac{A_j}{\hat{p}_j}} + \frac{\frac{1-A_i}{1-\hat{p}_i}}{\sum_{j=1}^{n} \frac{1-A_j}{1-\hat{p}_j}}$$

and we check for the balance of weights in the online and in-person groups across the Gender and International variables

```
# estimate the stabilized IPW weights
weights_alt_trt <- 1 / sum(trt / preds) * trt /preds
weights_alt_notrt <- 1 / sum((1 - trt)/(1 - preds)) * (1-trt)/(1-preds)
dat <- data.frame(dat, weights = weights_alt_trt - weights_alt_notrt)

# check balance for gender and international (other variables are also balanced)
dat %>% group_by(Gender, Online) %>% summarise(sum(weights))
```

```
## `summarise()` has grouped output by 'Gender'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 3
## # Groups:   Gender [2]
##   Gender Online `sum(weights)`
##    <int>  <int>          <dbl>
## 1      0      0         -0.563
```

```
## 2      0      1          0.574
## 3      1      0         -0.437
## 4      1      1          0.426
```

```
dat %>% group_by(International, Online) %>% summarise(sum(weights))
```

```
## `summarise()` has grouped output by 'International'. You can override using the
## `.groups` argument.
```

```
## # A tibble: 4 x 3
## # Groups:   International [2]
##   International Online `sum(weights)`
##           <int>  <int>          <dbl>
## 1             0      0         -0.587
## 2             0      1          0.596
## 3             1      0         -0.413
## 4             1      1          0.404
```

We now estimate the ATE corresponding to these weights.

```
ATE_alt <- sum(weights_alt_trt * dat$ObjExam) -
  sum(weights_alt_notrt * dat$ObjExam)
ATE_alt
```

```
## [1] 0.5775491
```

We now estimate the doubly robust ATE

```
# estimate DR version ATE
m_trt <- lm(ObjExam ~ ACTMath + ACTMajor + ACT + International + Gender +
            FR + SO + JR + F17 + S18 + S19,
            data = dat[trt == 1, ])
Y_trt <- predict(m_trt, newdata = dat)
m_notrt <- lm(ObjExam ~ ACTMath + ACTMajor + ACT + International + Gender +
              FR + SO + JR + F17 + S18 + S19,
              data = dat[trt == 0, ])
Y_notrt <- predict(m_notrt, newdata = dat)
ATE_DR <- mean( (dat$ObjExam * trt - (trt - preds) * Y_trt) / preds -
  (dat$ObjExam * (1 - trt) + (trt - preds)*Y_notrt) / (1 - preds))
ATE_DR
```

```
## [1] 0.4217033
```

The two quantities are essentially the same, both estimates of the ATE show that in-person instruction has no advantage over online instruction in STAT 200. **Construct a nonparametric bootstrap procedure that estimates the uncertainty associated with both estimates of the ATE. Do the conclusions change when we factor in the uncertainty obtained from the nonparametric bootstrap procedure? Explain.**

Acknowledgments

These notes borrow materials from Trevor Park's STAT 426 notes and Charles Geyer's notes on exponential families and other topics. We also borrow materials from Agresti [2013], Faraway [2016], and Sundberg [2019].

References

A. Agresti. *Categorical Data Analysis*. John Wiley & Sons, 2013.

Justin Esarey and Andrew Pierce. Assessing fit quality and testing for misspecification in binary-dependent variable models. *Political Analysis*, 20(4):480–500, 2012.

J. J. Faraway. *Extending the linear model with R: generalized linear, mixed effects and nonparametric regression models*. CRC press, 2016.

R. Sundberg. *Statistical Modelling by Exponential Families*, volume 12. Cambridge University Press, 2019.