Day 4 - Logistic Regression

Your Name

9/10/2021

if the libraries in this chunk don’t load properly, run the next chunk instead

if the above libraries loaded properly, there is no need to run this chunk

```
knitr::opts_chunk$set(echo = TRUE)
knitr::opts_chunk$set(fig.show = "hold")
knitr::opts_chunk$set(eval = FALSE)
remove.packages("rlang")
install.packages("rlang")
library(rlang)
library(faraway)
library(ggplot2)
library(dplyr)
```

Agenda

1. mathematical formulation.
    - motivation
    - Bernoulli + binomial
    - logistic and logit functions
    - odds ratio
2. logistic regression.
    - data investigation
    - performing a regression
    - inference
    - diagnostics
    - model selection
    - goodness-of-fit
3. linear separability.
4. machine learning application.

---

1. mathematical formulation

motivation

For the first part, we are going to look at data relating the rate of heart disease to smokers.

```
data(wcgs, package="faraway")     ##install the wcgs dataset from the faraway package

wcgs$y <- ifelse(wcgs$chd == "no",0,1)
#ifelse command operates as a shortcut of an if/else function

plot(jitter(y,0.1) ~ jitter(cigs), wcgs, xlab="# Cigs", ylab="Heart Disease", pch=".", main="Raw Data, Heart Disease vs. # Cigs Smoked")

idx1 <- wcgs$y == 0

idx2 <- wcgs$y == 1

plot(density(wcgs$cigs[idx1]), main="Outcome of Heart Disease against # Cigs smoked", xlab="# Cigs")

lines(density(wcgs$cigs[idx2]), lty='dashed')

legend("topright", c("Y=0", "Y=1"), lty=c('solid','dashed'), cex=1)
```

we see that we have two separate, but overlapping, density estimates for the rate of heart disease vs. smokers. linear regression runs into two major problems in problems such as these:

1. the response variable here is restricted to [0,1], whereas linear regression is unbounded in Y
2. the densities overlap significantly, making prediction of the *outcome* difficult–e.g., at 0 cigarettes, both densities overlap considerably, so if a patient doesn't smoke, by magnitude, we have to assume that they will not develop heart disease. this is not a useful metric.

logistic regression was developed to handle both of these problems elegantly. we will get to how shortly, but first we introduce some useful concepts.

what is a Bernoulli trial?

a Bernoulli trial is an experiment with two outcomes, say $Y = 0$ for a 'failure' and $Y = 1$ for a 'success', with probability p of success and $1 - p$ of failure. e.g., flipping a fair coin has $p = .5$ of heads ('success').

what is the Binomial distribution?

the binomial is an extension of the Bernoulli trial (or, if you prefer, the Bernoulli is a binomial with a single trial). more specifically, the binomial can be constructed from the sum of Bernoulli trials. with the coin flip, a single flip is a Bernoulli RV. let's suppose we turn it into a game, where you give your friend 10 chances (i.e. 10 coin flips) to get at least 7 heads; if they do, you'll buy them coffee. whereas each individual flip is a Bernoulli trial, one full game is a **binomial**.

more generally, define m as the total number of trials, each of which is independent from the rest (above, $m = 10$). each trial has the same probability of success, and we call that probability θ. define y to be the number of **successes** in m trials (e.g. if your friend played the game and got 6 heads, $y = 6$). then we can say that y has a binomial distribution with parameters m and θ; we write it $y \sim Bin(m, \theta)$.

the probability that y equals a specific integer j (e.g. $Pr(y = 6)$) is given by its **probability mass function**,

$$Pr(y = j) = \binom{m}{j}\theta^j(1-\theta)^{m-j}$$

where $\binom{m}{j} = \frac{m!}{j!(m-j)!}$. for our above example with the coins, the probability that your friend gets exactly 7 heads and at least 7 heads, respectively, are

```
dbinom(7, 10, .5)
pbinom(6, 10, .5, lower.tail=FALSE)
```

doing some math off-screen, we can get that the expectation and variance of the binomial distribution are

$$\mathbb{E}(y) = m\theta$$
$$\text{Var}(y) = m\theta(1-\theta)$$

notice that since m is a constant, both the expectation and the variance are entirely controlled by the θ parameter.

notably, this does **not** assume a constant value for θ, and in fact, this formulation allows us to define $\theta := \theta(\mathbf{x})$ – i.e., the parameter controlling the binomial distribution can be a function of a set of random variables $\mathbf{x}$!

logistic regression

as with regular regression, call Y our 'response variable' that can only take values of either 0 or 1, and let X be a set of predictor variables (also called covariates), where each Y_i has an associated set of $(X_{i1}, X_{i2}, \ldots, X_{in})$. the predictor variables can have any form, quantitative or qualitative, and can take any range of values, so long as they are linearly related; in other words, as long as we can express their relationship in the form

$$\mathbf{X}_i\boldsymbol{\beta} = \beta_0 + \beta_1 X_{i1} + \cdots + \beta_n X_{in}$$

there are an infinite number of functions $f : (-\infty, \infty) \mapsto (0, 1)$), and the one we will be using for logistic regression is called, believe it or not, the **logistic function**:

$$\theta(\mathbf{X_i}) = \eta(\mathbf{X_i}\boldsymbol{\beta}) = \frac{\exp(\mathbf{X_i}\boldsymbol{\beta})}{1+\exp(\mathbf{X_i}\boldsymbol{\beta})} = \frac{1}{1+\exp(-\mathbf{X_i}\boldsymbol{\beta})}$$

```
x <- seq(-10, 10, by=.1)
plot(x, ilogit(x), type="l", main="Logistic Function, CDF", ylab="cumulative probability")
```

we call η generally the 'link function', so in this case our 'link function' is the logistic function. it has an inverse called the **logit**,

$$\text{logit}(\theta(\mathbf{X_i})) = \log\left(\frac{\theta(\mathbf{X_i})}{1-\theta(\mathbf{X_i})}\right) = \mathbf{X_i}\boldsymbol{\beta}$$

```
x <- seq(.01, .99, by=.01)
plot(x, logit(x), type='l', main="logit function, PDF", xlab="probability")
```

as a note on jargon, we call $\mathbf{X_i}\boldsymbol{\beta}$ the **linear predictor**.

odds ratio

the odds ratio is a very common metric, especially in gambling circles; perhaps you've played the game odds with your friends? the ratio inside the logit function, $\frac{\theta(\mathbf{X_i})}{1-\theta(\mathbf{X_i})}$, is called the 'odds ratio' or 'odds of success'. if the probability of success is $\theta = .25$, then the odds ratio is 1:3, one success to three failures. alternatively, if the probability of success is .8, then the odds ratio is .8/.2 = 4:1, or four successes to one failure.

perhaps you've seen roulette table payout odds? e.g.,

- even/odd; black/red; low; high all have 1:1
- the dozens and columns all have 2:1
- six line is 5:1
- corner 8:1
- street 11:1
- split 17:1
- single 35:1

notice that these odds are the rates of success of the **house**, and are chosen to be slightly larger than the probability of each event happening (e.g. even/odd has 1:1 odds, or 50% chance of payout, but the chance of getting even or odd is only 48.6% due to 0 and 00 – the payout might be even, but the probability **always favors the house**).

the house always wins, but at least statisticians get good example problems out of it.

summary

we are interested in modeling a binary response variable Y using its predictors X. we let each Y_i be binomial with parameters $Y_i \sim \text{Bin}(m_i, \theta(X_i))$. because θ is bounded to $[0, 1]$, we have to transform our predictors X to fall inside those bounds. we do this by performing a linear regression where our response variable is the **logit** of θ, and our linear predictor is given by the standard linear regression formulation.

tl;dr: each response Y_i is modeled as a binomial distribution with a probability parameter determined by the values of its predictor X_i.

---

2. Performing a Logistic Regression

let's do a full, soup-to-nuts logistic regression on the 'wcgs' dataset from the faraway package. it was collected during the 'Western Collaborative Group Study', where about 3100 healthy men, ages 39-59, were assessed for their personality type; eight and half years later, information on blood pressure, cigarette consumption, and coronary heart disease were collected.

data investigation

let's go ahead and load the data; we'll start by looking at just a few of the variables:

```
data(wcgs, package="faraway")
summary(wcgs[,c("chd","height","cigs")])
```

let's go ahead and plot some of the data:

```
plot(height ~ chd, wcgs, main="Dist. of Height vs. Development of CHD")
wcgs$y <- ifelse(wcgs$chd == "no",0,1)
plot(jitter(y,0.1) ~ jitter(height), wcgs, xlab="Height", ylab="Heart Disease", pch=".", main="CHD vs. Height")
```

note the use of the `jitter()` function, which adds a tiny bit of noise to the data so that we don't overplot – gives us an idea of how many points there are at each 'point', since height is reported as a discrete value.

while we could do some intensive R formatting, we can also just use the `ggplot` library to make much more helpful graphs much more quickly:

```
ggplot(wcgs, aes(x=height, color=chd, fill=chd)) + geom_histogram(position="dodge", binwidth=1) + labs(title="CHD vs. Height")
ggplot(wcgs, aes(x=cigs, color=chd, fill=chd)) + geom_histogram(position="dodge", binwidth=5, aes(y=..density..)) + labs(title="CHD vs. # Cigs")

ggplot(wcgs, aes(x=height,y=cigs))+geom_point(alpha=0.2, position=position_jitter())+facet_grid(~ chd)
```

based on these graphs, why are we choosing logistic regression? well, we have two goals:

1. predict heart disease outcome for a given individual
2. explain the relationship between heart disease and the other variables

in this example, we observe that for the same height and cigarette usage, both outcomes occur, and both outcomes occur frequently. therefore, it makes better sense to model the **probability of getting heart disease** rather than the *rate of getting heart disease itself*. in other words, instead of predicting whether or not you have heart disease, we want to find out "what are the odds that you have heart disease, given that you're tall and smoke <#> cigarettes per day?"

performing a regression

we already developed the theory behind logistic regression; as a reminder,

$$\log\left(\frac{\theta_i}{1-\theta_i}\right) = \eta_i = \mathbf{X_i}\boldsymbol{\beta}$$

unlike OLS, there isn't a closed-form solution that allows us to solve for the optimal β; instead we use computational methods. the gist of it is that we use the Maximum Likelihood Estimate (MLE):

$$l(\boldsymbol{\beta}) = \sum_{i=1}^{n} \left[y_i\eta_i - \log(1+e^{\eta_i})\right]$$

we solve this iteratively to find the $\hat{\beta}$s and their associated SEs. of course, this is the 'royal we', as we let R take care of all that using `glm` ('general linear model'):

```
lmod <- glm(chd ~ height + cigs, family = binomial, wcgs)
summary(lmod)
```

let's investigate these coefficient values – how do we interpret them? let's start by varying the height and plotting contours for # of cigs smoked:

```
(beta <- coef(lmod))

plot(jitter(y,0.1) ~ jitter(height), wcgs, xlab="Height", ylab="Heart Disease",pch=".",
 main="Heart Disease vs. Height")
curve(ilogit(beta[1] + beta[2]*x + beta[3]*0),add=TRUE)
curve(ilogit(beta[1] + beta[2]*x + beta[3]*20),add=TRUE,lty=2)
legend("topleft", c("Nonsmoker", "Pack a Day"), lty=c('solid','dashed'), cex=.8)
```

we can then compare that to the opposite, varying # of cigs and plotting height contours:

```
plot(jitter(y,0.1) ~ jitter(cigs), wcgs, xlab="Cigarette Use", ylab="Heart Disease",pch=
".", main="CHD vs. # Cigs/Day")
curve(ilogit(beta[1] + beta[2]*60 + beta[3]*x),add=TRUE)
curve(ilogit(beta[1] + beta[2]*78 + beta[3]*x),add=TRUE,lty=2)
legend("topright", c('60in tall', "78in tall"), lty=c('solid','dashed'), cex=.8)
```

let’s return to our logistic model, and the idea of odds:

$$\log\left(\frac{\theta}{1-\theta}\right) = \log(\text{odds}) = \beta_0 + \beta_1 x_1 + \beta_2 x_2$$
$$\text{odds} = e^{\beta_0} e^{\beta_1 x_1} e^{\beta_2 x_2}$$

therefore, β_1 can be interpreted as follows: a one-unit increase in x_1 (i.e. height) will increase the odds of success (i.e. the odds of developing CHD) by a factor of e^{β_1}. therefore, the exponentiated coefficients will be more useful in the interpretation:

```
exp(beta)
```

therefore, the odds of developing CHD increase by 2.55% (remember, we *multiply* the odds by 1.0255) for each added inch in height *keeping # cigs the same*. similarly, the odds of developing CHD increase by 2.34% for each additional cigarette smoked per day, *controlling for height*.

sidenote: $\exp(x) \approx 1 + x$ for small x, and we see that $\beta_1 = .0252, \beta_2 = .231$ give very close appx. to our exponentiated coefficients. this is a good way to estimate the effect size when doing a rapid regression.

how much do our odds of developing CHD increase when we smoke a pack a day? well,

```
exp(beta[3]*20)
```

they increase by nearly 60%. **HOWEVER**: this regression is not nearly sufficient to develop any causation re:cigarettes causing CHD. just like with OLS, we can only conclude that the two are *associated*. there are ways to show causality; this is not one of them.

another note about interpretation: we have just calculated odds ratios. this means that we only have the odds relative to the predictor variables, we do not have magnitudes. smoking a pack a day **does not** mean you have a 60% chance of developing CHD; it means your odds of developing CHD if you smoke a pack a day are 60% higher than your odds of developing CHD if you don’t smoke. if the odds of a nonsmoker developing CHD is 1:99 (1%), the chance you develop CHD if you smoke a pack a day is only 2:123 (1.6%).

the **relative risk** is related to the odds ratio, and is the ratio of probabilities. we can calculate the probability by calculating the logistic function:

```
c(ilogit(sum(beta*c(1,68,20))),ilogit(sum(beta*c(1,68,0)))) # actual probabilities
ilogit(sum(beta*c(1,68,20)))/ilogit(sum(beta*c(1,68,0))) # relative risk
```

the relative risk of 1.54 is quite similar to the odds ratio of 1.59; for low probability outcomes, these two values will be similar, but this is certainly not the case for larger probabilities.

inference

we can use the likelihood ratio statistic to develop the ‘Deviance’, an extension of residuals that measure how well the data fit the model:

$$D = -2\sum_{i=1}^{n} \hat{p}_i \text{logit}(\hat{p}_i) + \log(1 - \hat{p}_i)$$

where $\hat{p}_i$ are the fitted values from the model. more broadly, this statistic can be used to test how well the model fits, but unfortunately, for math reasons, the logistic regression Deviance is just a function of fitted values and cannot be used for these purposes; we have to use other tests. however, we can still use the Deviance to compare two models to one another. recall the summary output:

```
summary(lmod)
```

the `Null Deviance` is a model with no predictors, just an intercept term. for math reasons, the difference between the Deviance of the larger model D_L and the Deviance of the smaller model D_S is χ^2_{L-S} distributed (at least asymptotically), with dof = difference in number of parameters between larger and smaller model (in our model, we add two predictors relative to the intercept-only model, so we have 2 dof):

```
1-pchisq(32.2,2)
```

this value is tiny, so we are confident there is a relationship between the predictors and the response. we can use the ANOVA to compare models to one another:

```
lmodc <- glm(chd ~ cigs, family = binomial, wcgs)
anova(lmodc,lmod, test="Chi")
```

it would seem that height does not play a significant role in predicting CHD when cigarettes are already accounted for, so we can drop it from our model. we can use the `drop1` function to test each predictor independently:

```
drop1(lmod,test="Chi")
```

this is a better alternative to the Z-test reported in the `summary()` window, but in this case they give us similar results. in many cases, especially those with sparse data (e.g. rare events), the standard errors get overestimated, shrinking the z-values and causing us to potentially miss out significant events. (nb: called the Hauck-Donner effect).

again for Hauck-Donner reasons, while we can use the asymptotic normality CIs,

$$\hat{\beta}_i \pm z^{\alpha/2} \text{se}(\hat{\beta}_i)$$

we prefer to use the likelihood-based CIs from `confint` :

```
confint(lmod)
```

diagnostics

as with OLS, our first step is to check the residuals (not deviances! not here). there are two types:

1. linear predictor scale, η
   - `predict(lmod)`
2. predicted probability, $\theta = \text{logit}^{-1}(\eta)$
   - `predict(lmod, type='response')`

we choose to use the 2nd, and calculate the **raw residuals**, plotting them against the fitted values:

```
linpred <- predict(lmod)
#equivalence: predprob <- ilogit(linpred)

predprob <- predict(lmod, type="response")
rawres <- wcgs$y - predprob
# equivalent: rawres <- residuals(lmod, type="response")

plot(rawres ~ linpred, xlab="linear predictor", ylab="residuals", main="Raw Resid. wrt L
inear Predictor")
```

(we only plotted vs. the linear predictor here because it gives us better, cleaner spacing than the probabilities. feel free to use either).

this is not a helpful chart, since our options are either 0 or 1. we don't expect normally distributed residuals, since our variance is $\theta(1-\theta)$, giving us highest variance near $\theta = .5$ and variances near 0 at either end. instead, let's take a look at the deviance residuals, $r_i = \text{sign}(y_i - \hat{p}_i)\sqrt{r_i^2}$, where $\text{Deviance} = \sum_i r_i^2$. these residuals are the default of `residuals(lmod)` .

this next part is a bit weird, but the gist of it is that we group residuals together into bins. each bin is based on similar predictor values, and the number of bins is chosen to match the size of the data. we choose 100 bins so that each bin has ~30 data points, and we will take advantage of the `dplyr` library:

```
wcgs <- mutate(wcgs, residuals=residuals(lmod), linpred=predict(lmod))
gdf <- group_by(wcgs, cut(linpred, breaks=unique(quantile(linpred,(1:100)/101)))) # make
bins

# get means of residuals and linear predictors
diagdf <- summarise(gdf, residuals=mean(residuals), linpred=mean(linpred))
plot(residuals ~ linpred, diagdf, xlab="linear predictor", main="Binned Residuals")
```

we also plot the binned residuals vs each of the predictors:

```
#height
gdf <- group_by(wcgs, height)
diagdf <- summarise(gdf, residuals=mean(residuals))
ggplot(diagdf, aes(x=height,y=residuals)) + geom_point()

#investigate the outlier bin
filter(wcgs, height==77) %>% select(height, cigs, chd, residuals)

#cigs
group_by(wcgs, cigs) %>% summarise(residuals=mean(residuals), count=n()) %>% ggplot(aes
(x=cigs, y=residuals, size=sqrt(count))) + geom_point()
```

we see occasionally ‘outlying’ points, but the bin sizes are quite small, and easily ignored. the main trend binned residuals are normally distributed around zero, indicating a good model.

let’s try and detect unusual points:

```
qqnorm(residuals(lmod))
```

unforunately, the Q-Q plot so useful in OLS is useless here; we see two obvious clusters, the lower one for $Y = 0$ and the higher one for $Y = 1$. instead, we will use the ‘half normal plot’ to vizualize our residuals:

```
halfnorm(hatvalues(lmod))
```

the gist of this plot is that we should see a continuous sequence of points; any ‘clustering’ behavior is an indication of points that are not following the trend. we see two such points above, and can filter them:

```
filter(wcgs, hatvalues(lmod) > 0.015) %>% select(height, cigs, chd)
```

we see that these two men smoke the largest number of cigarettes. these two points aren’t especially extreme, and we have a large enough dataset (~3100 points) that we can safely ignore them. it is more trouble than it’s worth to remove them from our dataset.

model selection

we will use ‘backward selection’ to find our model:

1. start with the full model, including all available predictors, and interaction terms if desired
2. compare this model with all models with one fewer predictor; compute the p-value for each dropped predictor (e.g. via `drop1`)
3. eliminate the predictor with the largest p-value greater than some threshhold (e.g. $p = .05$); if the criteria cannot be met, then stop and use this model. otherwise, return to step 2

HOWEVER: this is a horrible method to use, no matter how easy it is. while it is often used, it does not identify the best set of predictors for predicting future responses. it does not reliably indicate which predictors are the best explanation of the response. there is no ‘true’ model, and this can convince us (falsely) that there is. why? because the predictors are eliminated in a data-dependent way!!!

AIC

the AIC (Akaike Information Criterion) is a popular workaround to these problems, which uses likelihoods and number of parameters q: $\text{AIC} = -2 \log L + 2q \propto \text{Deviance} + 2q$. we select the model with the smallest AIC; when the AIC stops shrinking, we stop trying new models.

```
# combine height and weight into BMI
wcgs$bmi <- with(wcgs, 703*wcgs$weight/(wcgs$height^2))

lmod <- glm(chd ~ age + height + weight +bmi + sdp + dbp + chol + dibep + cigs +arcus, f
amily=binomial, wcgs)
lmodr <- step(lmod, trace=0)
summary(lmodr)
```

we have dropped only two regressors: weight and diastolic blood pressure.

another caveat: AIC works great for *predictive* purposes, and is one method among several that excel in that arena. however, notice that we have eliminated the diastolic blood pressure as a predictor of having CHD – we may be tempted to assume that dBP has no relation to the chance of heart disease. well, try regressing on it alone:

```
drop1(glm(chd ~ dbp, family=binomial, wcgs), test="Chi")
```

this assumption turns out to be quite false; dBP is related to CHD. we should not be so bold to assume that our model selection methods are choosing **causal**, or even **correlated**, predictors for CHD in terms of risk factors. this is not a medical/scientific analysis! there isn't an 'experiment' we are investigating. we are simply uncovering relationships between CHD and the measured predictors. model selection is not a stand-in for determining risk factors, let alone causative inference.

goodness-of-fit

we will not discuss most goodness-of-fit methods, as they are complicated and not really that interesting, imo. but we can discuss scoring methods:

we create a 2x2 table for classification by the model vs. true classification:

```
wcgsm <- na.omit(wcgs)
wcgsm <- mutate(wcgsm, predprob=predict(lmod,type="response"))
wcgsm <- mutate(wcgsm, predout=ifelse(predprob < 0.5, "no", "yes"))
xtabs( ~ chd + predout, wcgsm)

#correct classification rate:
(2882+2)/(2882+3+253+2)

#1 - false positive rate, specificity
2882/(2882 + 3)

#1 - false negative rate, sensitivity
2/(253 + 2)
```

```
thresh <- seq(0.01,0.5,0.01)
Sensitivity <- numeric(length(thresh))
Specificity <- numeric(length(thresh))
for(j in seq(along=thresh)){
  pp <- ifelse(wcgsm$predprob < thresh[j],"no","yes")
  xx <- xtabs( ~ chd + pp, wcgsm)
  Specificity[j] <- xx[1,1]/(xx[1,1]+xx[1,2])
  Sensitivity[j] <- xx[2,2]/(xx[2,1]+xx[2,2])
}

matplot(thresh,cbind(Sensitivity,Specificity),type="l",xlab="Threshold",ylab="Proportion",lty=1:2, main="Sensitivity vs Specificity, fxn of threshold")
legend("bottomleft", c("Sensitivity", "Specificity"), lty=c('solid', 'dashed'), col=c('black', 'red'), cex=.8)
```

```
plot(1-Specificity,Sensitivity,type="l", main="ROC")
abline(0,1,lty=2)
```

3. Linear Separability

let’s take a look at the famous iris dataset, the one we looked at two lessons ago, again:

```
irisr <- filter(iris, Species != "virginica") %>%  select(Sepal.Width, Sepal.Length,Species)
(p <- ggplot(irisr, aes(x=Sepal.Width, y=Sepal.Length, shape=Species)) +geom_point())
```

we go ahead and perform a logistic regression, to see if we can separate the two species using the sepal dimensions:

```
lmod <- glm(Species ~ Sepal.Width + Sepal.Length, family=binomial, irisr)
summary(lmod)
```

and it…fails?

it turns out the residual deviance is 0, indicating a perfect fit. zero deviance, however, blows up the standard errors to infinity, hence the error and meaningless coefficients. when we can perfectly bisect our data, we say that the data is “linearly separable”, which makes interpretation impossible. it almost implies that we can perform perfect predictions on new data (which is a Holy Grail – desirable beyond all things, but not real), and we can’t quantify the certainty of our predictions because the standard errors are nonsense. we must resort to other methods.

there are several schools of thought, some of which use souped-up versions of logistic regression to perform “exact logistic regressions” (see the package `elrm`). we will use a bias-reduction method, implemented in the `brglm` library:

```
library(brglm)
bmod <- brglm(Species ~ Sepal.Width + Sepal.Length, family=binomial, irisr)
summary(bmod)

#plotting
p + geom_abline(intercept=(0.5+24.51)/9.73, slope=8.9/9.73)
```