Homework 5

Neal Marquez

November 27, 2018

```
rm(list=ls())
library(MASS)
library(nnet)
library(tidyverse)
library(mvtnorm)
library(latex2exp)
library(knitr)
library(arm)

# Load in the anes data set
anesDF <- "https://faculty.washington.edu/cadolph/mle/nes92con.csv" %>%
    read_csv %>%
    mutate(vote92=factor(c("Bush", "Clinton", "Perot")[vote92+1]))
```

Problem 1: Modeling Vote Choice with Multinomial Logit

In this problem you will again use data from the 1992 American National Election Study. The data are taken from a nationally representative sample of the US population. The survey was conducted in the fall of 1992 with a reinterview of the respondents following the election. In the data you will be using, only the vote report is taken from the post-election interview.

The dataset `nes92.csv` contains information on the vote for President and approval of then-President Bush along with several variables selected to capture the range of influences that current work on voting behavior expects to be important. Descriptions of the variables are at the end of this assignment.

A

Use a multinomial logit model to explain respondents' 1992 Presidential vote choice. Using `multinom()` from the `nnet` library, fit the model to the variable `vote92` with `rlibcon`, `rbdist`, `econ`, `gulfwar`, and `nonwhite` as the only covariates. Report the estimated parameters, their standard errors, and the value of the log likelihood at its maximum.

```
nnFit <- multinom(
    vote92 ~ rlibcon + rbdist + econ + gulfwar + nonwhite,
    data = anesDF,
    trace = FALSE)
```

```
## Log Likelihood: -353.59
```

Candidate	Coefficient	Estimate	Std. Errors
Clinton	(Intercept)	-1.1340	0.9935
Perot	(Intercept)	-2.1462	0.9890
Clinton	rlibcon	-0.5250	0.1101
Perot	rlibcon	-0.0765	0.1126
Clinton	rbdist	0.7920	0.1324
Perot	rbdist	0.6818	0.1342
Clinton	econ	0.5970	0.1712

Candidate	Coefficient	Estimate	Std. Errors
Perot	econ	0.2919	0.1587
Clinton	gulfwar	-0.7615	0.3134
Perot	gulfwar	-0.3114	0.3277
Clinton	nonwhite	0.7537	0.5165
Perot	nonwhite	-0.9946	0.7195

B

Calculate the probability that a white respondent voted for Bush, Clinton, or Perot, given each self-reported position on the liberal-conservative continuum (`rlibcon` $= \{1, 2, 3, 4, 5, 6, 7\}$ and all other covariates held at their means. Next, calculate the probabilities for a nonwhite respondent.

Below we build functions for calculating the probabilities and uncertainty via simulation for average covariate values and values of `rlibcon` described above for both white and nonwhite respondents.

```
# function for predicting
ratio2Pred <- function(data, model, draws=NULL, X=NULL){
    if(is.null(model$par)){
        model$par <- c(t(coefficients(model)))
    }
    G <- nrow(coefficients(model)) + 1
    if(is.null(X)){
        # lets not redo this for every draw
        X <- model.matrix(as.formula(model$call$formula[-2]), data)
    }
    if(!is.null(draws)){
        # recursive process for simulations
        betas_ <- rmvnorm(draws, model$par, vcov(model))
        y_prob <- array(0, dim=c(nrow(data), G, draws))
        for(d in 1:draws){
            m <- model
            m$par <- betas_[d,] # treat the sims like true values
            y_prob[,,d] <- ratio2Pred(data, m, X=X) # rerun function with sim
        }
    }
    else{
        # transform vector of beta paramters into matrix for ease of use
        betas_ <- matrix(model$par, ncol=G-1, nrow=ncol(X))
        # get the ratios for the groups vs ref
        ratio_ref <- exp(X %*% betas_)
        # the response var is transformed to probability space
        p1 <- apply(ratio_ref, 1, function(x) 1 / (1 + sum(x)))
        y_prob <- unname(cbind(p1, ratio_ref * p1))
    }
    return(y_prob)
}

predDF <- anesDF %>%
    select(rbdist, econ, gulfwar) %>%
    summarize_all(mean, na.rm=TRUE) %>%
    cbind(expand.grid(rlibcon=1:7, nonwhite=0:1))

nDraws <- 1000
```

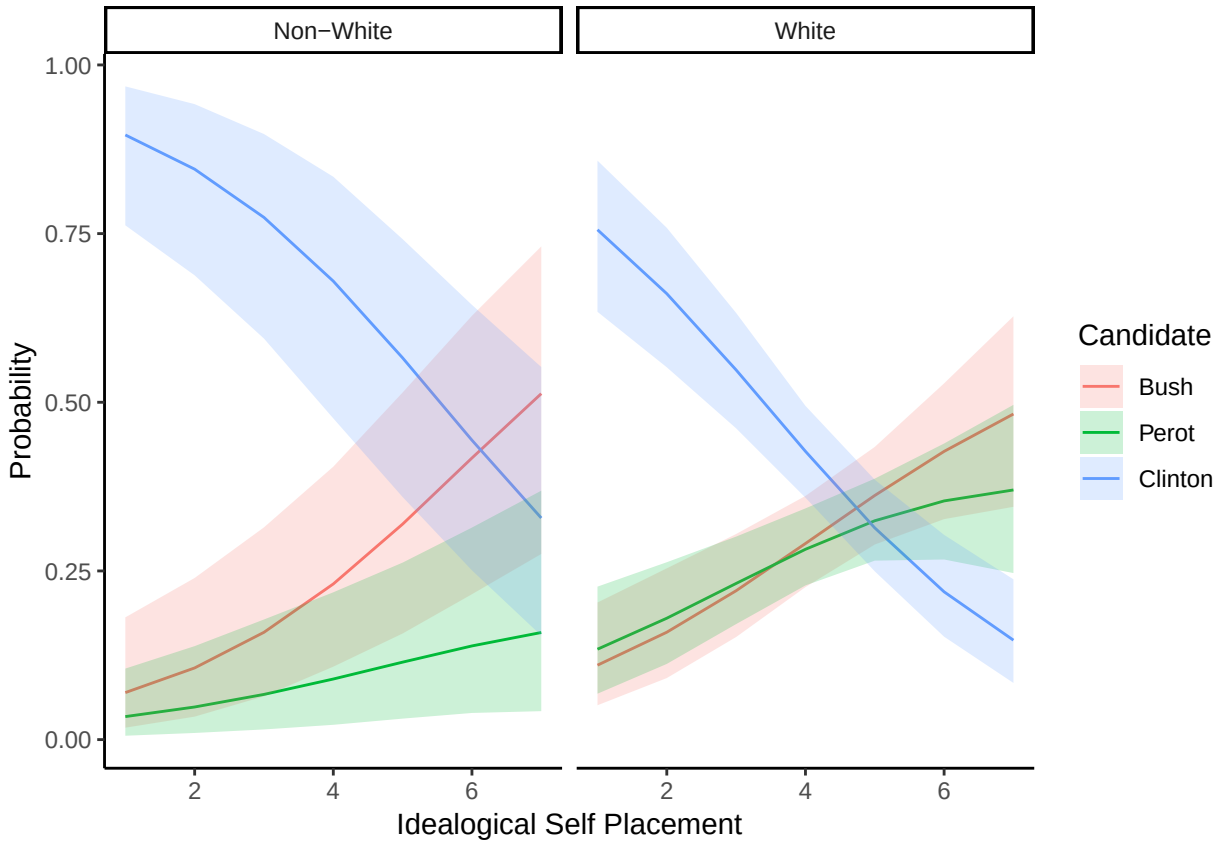
```
simArray <- ratio2Pred(predDF, nnFit, nDraws)
candidates <- levels(anesDF$vote92)
```

C

Present the expected probabilities of a vote for each presidential candidate calculated in part b in a way that would not require your reader to know anything about multinomial logit models. Include the uncertainty of your estimates in your presentation. Briefly discuss the results of the model.

The results below show the change in predicted probabilities across the spectrum of ideological placement where 1 is most liberal and 7 is most conservative. The results show what is probably obvious to most, that as ideology tends towards more conservative the probability of voting for Clinton decreases and the probability of voting for Perot and Bush increases. Interestingly, this effect is different for whites and non-whites, where nonwhites are more likely to vote for Clinton at all levels of political ideology.

```
bind_rows(lapply(1:nDraws, function(d){
    bind_rows(lapply(candidates, function(x) mutate(predDF, vote=x))) %>%
        mutate(draw=d)})) %>%
    mutate(value=c(simArray)) %>%
    group_by(rlibcon, vote, nonwhite) %>%
    summarize(
        prob = mean(value),
        plwr = quantile(value, probs=.025),
        pupr = quantile(value, probs=.975)) %>%
    ungroup %>%
    mutate(vote=factor(vote, levels=c("Bush", "Perot", "Clinton"))) %>%
    mutate(nonwhite=c("White", "Non-White")[nonwhite+1]) %>%
    ggplot(aes(x=rlibcon, y=prob, group=vote, ymin=plwr, ymax=pupr)) +
    geom_line(aes(color=vote)) +
    geom_ribbon(aes(fill=vote), alpha=.2) +
    theme_classic() +
    facet_wrap(~nonwhite) +
    labs(color="Candidate", fill="Candidate",
         x="Idealogical Self Placement", y="Probability")
```



Problem 2

A

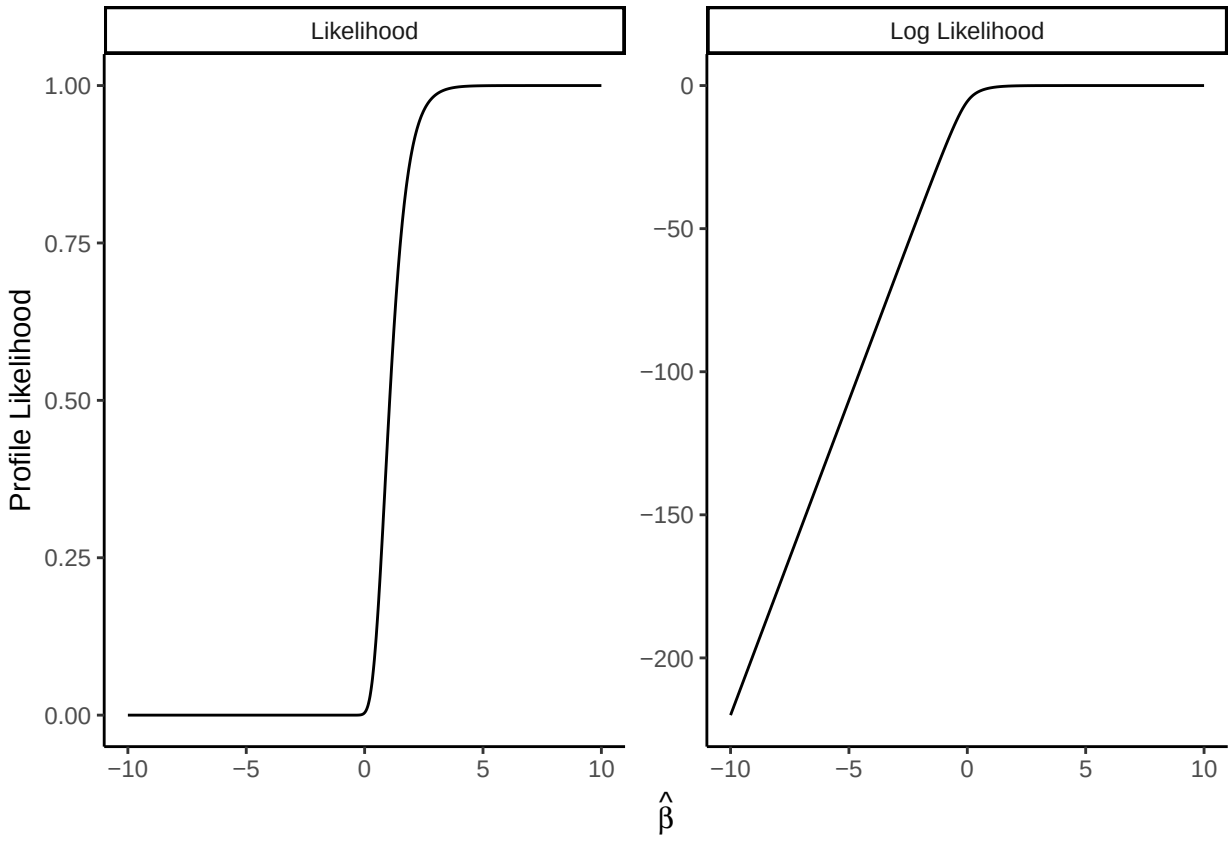
Consider two variables x and y where y is generated from x as a a Bernoulli draw with probability of success $\text{logit}^{-1}(x\beta)$ where the true value of β is 2 and is unknown from our perspective. Suppose that we observe eight instances of these variables, $\boldsymbol{y} = (1, 0, 0, 0, 1, 0, 0, 0)$ and $\boldsymbol{x} = (5, -2, -2, -2, 5, -2, -2, -2)$. Using the Bernoulli-logit likelihood, show the profile likelihood of β over the domain $[-10, 10]$. Interpret the likelihood surface you have found.

Whereas, we would expect to find a profile likelihood peak at a value and then decay at either side, the likelihood, and log-likelihood, continue to increase as the value increases. This is because the likelihood will continue to improve as $\hat{\beta}$ increases because the outcome is linearly separable using the covariates provided.

```
sampDF <- tibble(
    y = c(1,0,0,0,1,0,0,0),
    x = c(5, -2, -2, -2, 5, -2, -2, -2))

tibble(bhat=seq(-10, 10, length.out=1000)) %>%
    mutate(Likelihood=sapply(bhat, function(b){
        prod(dbinom(sampDF$y, 1, invlogit(b * sampDF$x)))
    })) %>%
    mutate(`Log Likelihood`=sapply(bhat, function(b){
        sum(dbinom(sampDF$y, 1, invlogit(b * sampDF$x), log=T))
    })) %>%
    gather(key="Type", value="lik", -bhat) %>%
    ggplot(aes(x=bhat, y=lik)) +
```

```
    geom_line() +
    theme_classic() +
    labs(x=TeX("$\\hat{\\beta}$"), y="Profile Likelihood") +
    facet_wrap(~Type, scales="free")
```



B

Estimate a logit model of y as a function of x and report the estimated coefficients and their standard errors. Use both the logit MLE and GLM with a logit link to estimate the model. Do you trust these results? What do you think is happening?

The results below show that for both GLM and MLE, the standard errors are exceedingly large and we should call into question whether these results are valid. What is likely happening is that the profile likelihood surface is extremely flat, which is expected as the likelihood function provides marginally larger values when we increase $\hat{\beta}$. This leads to huge estimates of the 95% confidence intervals.

```
glmFit <- glm(y ~ x + 0, data=sampDF, family=binomial)

binom1eval <- function(param){
    -sum(dbinom(sampDF$y, 1, invlogit(param * sampDF$x), log=TRUE))
}

optimFit <- optim(0, binom1eval, method="BFGS", hessian=TRUE)
st.err <- sqrt(optimFit$hessian^-1)

tibble(
    Estimate = c(coefficients(glmFit), optimFit$par),
```

```
    `Standard Error` = c(sqrt(vcov(glmFit)), st.err),
    Method = c("GLM", "MLE")) %>%
    kable(digits=2)
```

Estimate	Standard Error	Method
12.28	26577.17	GLM
11.00	12221.75	MLE

C

Now consider some real-world data, provided by Daniel Stegmuller. The file `India.Rdata` contains a dataframe called `govcollapse` which records incidents of government collapse in India (see table below for a description). Fit a logistic regression to the variable `collapse` using all the covariates in `govcollapse`. Report the estimated parameters, their standard errors, and the log likelihood at its maximum using `glm()`. Do you have any concerns about these estimates?

Again, we find a similar situation where the standard errors for the model are extremely large. This is because again the covariates provide a space where the response variable can be linearly separated, which leads to flat profile likelihoods and extreme standard errors. We should be very skeptical of the results provided by this model fit.

```
load(url("https://faculty.washington.edu/cadolph/mle/India.Rdata"))
govFit <- glm(collapse ~ . , data=govcollapse, family=binomial)
cat(paste0("Log Likelihood: ", signif(logLik(govFit), 5)))
```

```
## Log Likelihood: -2.9419e-10
```

```
summary(govFit)$coefficients %>%
    kable(digits=4)
```

	Estimate	Std. Error	z value	Pr(>|z|)
(Intercept)	-123.1447	441282.91	-3e-04	0.9998
iou	-0.1882	32684.99	0e+00	1.0000
ideology	0.4862	92476.73	0e+00	1.0000
splinter	0.1736	104241.38	0e+00	1.0000
opp	-0.2060	30026.85	0e+00	1.0000
extreme	0.2225	169868.30	0e+00	1.0000
govt_end	50.3630	82900.66	6e-04	0.9995

D

Examine the in-sample predicted probabilities of government collapse and plot them against each covariate. Do you notice any problems with these predictions? What might you do to improve them and would your proposed fix have any downside? Does this example raise general concerns about the proper use of logistic regression?

The predictions below show that across covariates we have nearly certain predictions of 1 or 0. In order to improve our predictions we should trim down the number of covariates that we use and test our model in terms of out of sample validity. A downside to this is that the inclusion of all variables may in fact be the true data generating process but the data that we have does not permit us to use an MLE approach to fit the model. This example shows us the limitations that we reach when fitting a model with an MLE approach and we may need to use a Bayesian approach if we truely believe the inclusion of all variables best reflects the process being analyzed.

```
govcollapse %>%
    mutate(pred=invlogit(predict(govFit))) %>%
    select(-collapse) %>%
    gather("Variable", "Value", -pred) %>%
    ggplot(aes(Value, pred)) +
    geom_point() +
    theme_classic() +
    facet_wrap(~Variable) +
    labs(x="Covariate Value", y="Prediction")
```