Simulation

Simulation and Model Assessment

William Reed

Using simulation to learn about the distribution of quantities of interest $q(\beta, X_1, X_2, ...)$.

1. Estimate the model to obtain $\hat{\beta}$ and $\hat{\Sigma}$.
2. Choose the row vectors X_1, X_2,...
3. Do the following $n_{sims} \geq 1000$ times:
   3.1 Simulate a parameter vector $\tilde{\beta} \sim N(\hat{\beta}, \hat{\Sigma})$.
   3.2 Calculate $\tilde{q}_i = q(\beta, X_1, X_2, ...)$.
4. Characterize the distribution of $\tilde{q}$.
   4.1 mean, median, mode
   4.2 standard deviation
   4.3 quantiles ([5th, 95th] $\rightarrow$ 90% CI; [2.5th, 97.5th] $\rightarrow$ 95% CI)

```
> # load libraries
>> library(arm)
> n <- 1000
> educate <- rnorm(n)
> age <- rnorm(n)
> income<-rnorm(n)
> X <- cbind(1,educate,age,income)
> b <- c(1,1,-2,2)
> p <- plogis(X%*%b)
> vote <- rbinom(n, 1, p)
> # estimate simple logit model
> m <- glm(vote ~ educate + age + income, family = binomial)
> # display results
> display(m, detail = TRUE)
glm(formula = vote ~ educate + age + income, family = binomial)
coef.est coef.se z value Pr(>|z|)
(Intercept)   0.97     0.11    9.18    0.00
educate       1.00     0.11    9.35    0.00
age          -1.93     0.15  -13.27    0.00
income        2.01     0.15   13.64    0.00
---
n = 1000, k = 4
residual deviance = 691.6, null deviance = 1326.2 (difference = 634.6)
```

```
# set the medians
x.edu <- median(educate)  # in years
x.age <- median(age)  # in years
x.inc <- median(income)  # in 10,000s of dollars

# set beta.hat, Sigma.hat, and X.c
beta.hat <- coef(m)
Sigma.hat <- vcov(m)
X.c <- c(1, x.edu, x.age, x.inc)

# simulate p.tilde
library(MASS)  # mvrnorm()
n.sims <- 1000  # set number of simulations
p.tilde <- numeric(n.sims)  # create holder for simulations
for (i in 1:n.sims) {
        beta.tilde <- mvrnorm(1, beta.hat, Sigma.hat)
        p.tilde[i] <- plogis(X.c%*%beta.tilde)
}

# characterize the simulations
mean(p.tilde)
sd(p.tilde)
quantile(p.tilde, c(0.05, 0.95))
```

```
vote<-data.frame(vote,educate,age,income)

mod <- obsval(vote~educate+age+income,data=vote,
reg.model = "logit",
n.draws = 1000,
effect.var = "age",
effect.vals = c(0,2), # lowest to mid
verbose = TRUE)

# display model results
summary(mod$model)
summary(mod$preds)
# see the names of everything obsval returns
# names(mod)
```

```
vioplot(mod$sim.coef[,1], mod$sim.coef[,2], mod$sim.coef[,3],
mod$sim.coefs[,4], names=c(expression(beta[0]),expression(beta[age]),
expression(beta[educate]),expression(beta[income])),
col="red")
title("Coefficients")

vioplot(mod$preds[,1], mod$preds[,2], names=c("age=0","age=2"),
col="red")
title("Predicted Probability")
mean(mod$preds[, 1] - mod$preds[, 2])
quantile(mod$preds[, 1] - mod$preds[, 2], c(0.025, 0.975))
```

```
> library(lmtest)
> n <- 1000
> educate <- rnorm(n)
> age <- rnorm(n)
> income<-rnorm(n)
> X <- cbind(1,educate,age,income)
> b <- c(1,1,-2,2)
> p <- plogis(X%*%b)
> vote <- rbinom(n, 1, p)
> # estimate simple logit model
> m <- glm(vote ~ educate + age + income, family = bino
> m0<-glm(vote~1, family=binomial)
> # display results
> display(m, detail = TRUE)
```

```
glm(formula = vote ~ educate + age + income, family = binomial)
coef.est coef.se z value Pr(>|z|)
(Intercept)   1.05     0.11    9.51    0.00
educate       0.90     0.11    8.16    0.00
age          -2.13     0.16  -13.64    0.00
income        2.25     0.16   13.85    0.00
---
n = 1000, k = 4
residual deviance = 665.2, null deviance = 1331.0 (difference =
> m

Call:  glm(formula = vote ~ educate + age + income, family = bi

Coefficients:
(Intercept)      educate          age       income
1.0453       0.8976      -2.1300       2.2479

Degrees of Freedom: 999 Total (i.e. Null);  996 Residual
Null Deviance:              1331
Residual Deviance: 665.2         AIC: 673.2
```

```
#Deviances
nd<-2*(0 - logLik(m0))
nd
rd<-2*(0-logLik(m))
rd
nd-rd


#Akaikes Information Criterion
length(coef(m))
AIC<- -2*(logLik(m))+2*(length(coef(m)))
AIC
#Bayesian Information Criterion
BIC<- -2*logLik(m)+log(n)*(length(coef(m))+1)
BIC

#Likelihood Ratio Tests
logLik(m0)
logLik(m)
lr.test = -2*(logLik(m0) - logLik(m))
lr.test
lrtest(m0,m)
```

```
#Things go bad

x<-c(0,0,0,0,0,1,1,1,1,1)
y<-c(0,0,0,0,0,0,0,0,1,1)


m1<-glm(y~x+z,family=binomial)

display(m1, detail = TRUE)
```