Lab 8: Multinomial Logit

Haohan Chen

March 2, 2018

Packages for Ordered or Unordered Outcomes

- VGAM: for ordered and unordered outcome
- polr: for ordered outcome
- mclogit: For conditional logit (by a political scientist)
- mlogit: multinomial logit model

This isn't really a authoritative package for this family of model. And different models have slight differences in their ways of specifying the models. Also they have some strict requirement about the input data (we all experienced that in HW3). So my advice would be using these packages with caution.

Conditional Logit Model

The Model Construct (Recitation)

R Implementation

```r
rm(list=ls())
library(mlogit)
load("BESdata.Rdata")
BESdata$id <- 1:nrow(BESdata)
identifiers <- data.frame(id = rep(1:nrow(BESdata), times=1, each=3),
                          candidate = rep(levels(BESdata$vote),
                                          times=nrow(BESdata), each=1))
d_t <- merge(identifiers, BESdata, by="id")
d_t$approval <- apply(d_t, 1, function(x) x[paste0("app", x["candidate"])])
d_t$approval <- as.numeric(d_t$approval)
d_t$choice <- (d_t$candidate == d_t$vote) + 0
# Transform the model into mlogit input
d_final <- mlogit.data(d_t, shape = "long", choice = "choice",
                    alt.var = "candidate", id = "id")
```

```r
# Fit a model
# ---------------
fit_clogit <- mlogit(choice ~ approval | union + gender + age, data = d_final)
summary(fit_clogit)

# Simulation of betas
# --------------------
n_sim <- 1000
beta_m <- fit_clogit$coefficients
beta_fisher_info <- solve(-fit_clogit$hessian)
beta_sim <- MASS::mvrnorm(n_sim, mu = beta_m, Sigma = beta_fisher_info)
rownames(beta_sim) <- paste0("sim", 1:n_sim)
```

```r
# Hypothesis testing
# --------------------
beta_ci <- as.data.frame(t(apply(beta_sim, 2, function(x) quantile(x, c(0.025, 0.5, 0.975)))))
names(beta_ci) <- c("beta_lo", "beta_m", "beta_hi")
beta_ci$var <- rownames(beta_ci)
# Plot your estimated coefficients and confidence itnerval
library(ggplot2)
ggplot(beta_ci, aes(x=var, y=beta_m)) + geom_point() +
  geom_errorbar(aes(ymin=beta_lo, ymax=beta_hi), width=0.2) +
  geom_hline(yintercept = 0, linetype="dashed", color="red") +
  ylab("Coefficients (95% Confidence Interval)") + xlab("Variable") +
  coord_flip()

head(beta_sim)
colnames(beta_sim)
# What can we say about the effect of approval on vote choice?
```

```r
# Predictive Probability
# ----------------------
# Construct a matrix of x. We want 10 levels of approval 1 to 10
# And fix the other covariates
colnames(beta_sim)

x_brown <- rbind(0, 0, 1:10, 0, 0, 0, 0, 0, 0)
```

```r
x_cameron <- rbind(1, 0, 5, 1, 0, 1, 0, 1, 0)
x_clegg <- rbind(0, 1, 5, 0, 1, 0, 1, 0, 1)
rownames(x_brown) <- rownames(x_cameron) <- rownames(x_clegg) <- colnames(beta_sim)
colnames(x_brown) <- colnames(x_cameron) <- colnames(x_clegg) <- paste0(as.character(1:10))
# Check what the x's look like
x_brown
x_cameron

# Calculate the odds
odds_brown <- beta_sim %*% x_brown
odds_cameron <- beta_sim %*% x_cameron
odds_clegg <- beta_sim %*% x_clegg

# Get the denominator for the predicted probability
sum_odds <- odds_brown + odds_cameron + odds_clegg
head(sum_odds)

# Calculate the simulated predicted probability
pred_prob_brown <- exp(odds_brown) / sum_odds
pred_prob_cameron <- exp(odds_brown) / sum_odds
pred_prob_clegg <- exp(odds_clegg) / sum_odds
head(pred_prob_cameron)
```

```r
# Plot them (in separate plots)
## Spaghetti plot
pred_prob_brown2 <- reshape2::melt(as.matrix(pred_prob_brown))
head(pred_prob_brown2)
pred_prob_brown2$Var2 <- as.numeric(pred_prob_brown2$Var2)
colnames(pred_prob_brown2) <- c("n_sim", "Approval", "Predicted Probability")
ggplot(pred_prob_brown2, aes(x=Approval, y=`Predicted Probability`, group=n_sim)) + geom_line(alpha=.1)
```

```r
# CI plot
pred_prob_brown_ci <- apply(odds_brown, 2, function(x) quantile(x, c(.025, .5, .975)))
```