STA5003: Categorical Data Analysis

Gong Wenwu 12031299

June 9, 2021

1. Poisson Log-linear model for rate

- **Q1-2.** See the Appendix.
- **Q3.** Card trick: 0.8823529, 0.7000000, 0.8000000, 1.0000000; Coin trick: 0.5416667, 0.6871795, 0.9134615, 0.3400000.
- **Q4.** Odds ratio: 7.764706. The CI is: (3.4404 , 17.5241).
- **Q5-6.** See the Appendix.

2. Models for Multinomial Responses (Loglinear Model)

- **Q1-2: Nominal Responses under $J = 3$.**
  Let $\pi_j(\boldsymbol{x}) = P(Y = j \mid \boldsymbol{x})$ at a fixed setting $\boldsymbol{x}$ for explanatory variables, with $\sum_j \pi_j(\boldsymbol{x}) = 1$. For observations at that setting, we treat the counts at the J categories of Y as a multinomial variate with probabilities $\{\pi_1(\boldsymbol{x}), \ldots, \pi_J(x)\}$. Logistic models pair each response category with a baseline category, such as the last one or the most common one. Consider the model

$$\log \frac{\pi_j(\boldsymbol{x})}{\pi_J(\boldsymbol{x})} = \alpha_j + \boldsymbol{\beta}_j^T \boldsymbol{x}, \quad j = 1, \ldots, J-1$$

  The left-hand side is the logit of a conditional probability, $\text{logit}[P(Y = j \mid Y = j \text{ or } Y = J)]$. This model simultaneously describes the effects of x on these $J - 1$ logits. The effects vary according to the response paired with the baseline.

  The G^2 value for reduced models indicate that both Gender and Race have effects and the baseline model (G+R) has $G^2 = 6.07475$ (df=2). We can say that the fit seems adequate by using the fitted values for model (G+R).

  Further, with YES as the baseline category, we can obtain ML estimates of effect parameters. We use letter subscripts to denote the belief categories, for example, the prediction equation

	Yes	Unsure	No
Female Black	92.3	10.9	2.8
Female White	49.7	12.1	4.2
Male Black	392.7	146.1	23.2
Male White	239.3	183.9	39.8

Figure 1: Model fitted values

for the log odds of selecting NO instead of YES is

$$\log\left(\hat{\pi}_{NO}/\hat{\pi}_{YES}\right) = -1.794 - 1.033G - 0.673R,$$

Gender and Race have a noticeable negative effect. For a given Gender, for Black people the estimated odds that belief choice is NO instead of YES are $\exp(-0.673) = 0.5$ times the estimated odds for White people. For a given Race, for female the estimated odds that belief choice is NO instead of YES are $\exp(-1.034) = 0.36$ times the estimated odds for Male.

Viewing all these, both gender and race have significant effects. The logistic model with additive effects and no interaction fits well, with $G^2 = 6.07475$ (df=2). The hypothesis of conditional independence of Gender and Belief, controlling for Race, is $H_0 : \beta_1 = 0$. The likelihood-ratio statistic equals 29.74203 (df = 2), showing evidence of association ($P < 0.001$). Same with Gender, Race is also dependent with Belief (LR=40.73175 with df = 2).

- **Q3: Ordinal Responses under $J = 3$.**
  The category ordering by forming logits of cumulative probabilities are

$$P(Y \leq j \mid x) = \pi_1(x) + \cdots + \pi_j(x), \quad j = 1, \ldots, J.$$

The we define cumulative logits are defined as

$$\begin{aligned}\text{logit}[P(Y \leq j \mid x)] &= \log \frac{P(Y \leq j \mid x)}{1 - P(Y \leq j \mid x)} \\ &= \text{logit}[P(Y \leq j \mid x)] = \alpha_j + \boldsymbol{\beta}^T \boldsymbol{x}, \quad j = 1, \ldots, J-1.\end{aligned}$$

The cumulative probit model is the cumulative link model using the standard normal cdf Φ for G. This generalizes the binary probit model to ordinal responses. It is appropriate when the conditional distribution for the latent variable Y^* is normal. Parameters in probit models refer to effects on $E\left(Y^*\right)$. For instance, consider the model $\Phi^{-1}[P(Y \leq j)] = \alpha_j - \beta x$. Since $Y^* = \beta x + \epsilon$ where $\epsilon \sim N(0, 1)$ has cdf Φ, a 1 -unit increase in x corresponds to a β increase in $E\left(Y^*\right)$. When ε need not be in standard form with $\sigma = 1$, a 1-unit increase in

x corresponds to a β standard deviation increase in $E(Y^*)$.

Under the belief NO is the reference category, the main-effects cumulative logit model of proportional odds form is

$$\text{logit}[P(Y \leq j \mid x)] = 0.07631 + 2.32238 + 0.76956G + 1.01645R$$

The parameter estimates yield estimated logits and hence estimates of $P(Y \leq j), P(Y > j)$, or $P(Y = j)$. We illustrate for White Female subjects $(R = 0)$, since $\hat{\alpha}_1 = 0.07631$, the estimated probability of response Belief YES is

$$\hat{P}(Y = YES) = \hat{P}(Y \leq YES) = \frac{\exp(0.07631 + 0.76956)}{1 + \exp(0.07631 + 0.76956)} = 0.6997001$$

Further, the estimated probability of response Belief Unsure is 0.2568589 and Belief NO is 0.043441. The effect estimates $\hat{\beta}_G = 0.76956$ and $\hat{\beta}_R = 1.01645$ suggest that the cumulative probability of Belief YES is higher for Blacks than for Whites and higher for Female than for Male. For example, given the Male, for Whites the estimated odds of reporting being belief YES were $e^{0.76956} = 2.158827$ times the estimated odds for Blacks. This estimate is imprecise, because relatively few observations were in the Black category.

Under the belief NO is the reference category, the corresponding cumulative probit model has fit

$$\Phi^{-1}[\hat{P}(Y \leq j)] = -0.06776 - 1.30293 - 0.44936G - 0.54371R$$

The nature of the effects and the substantive significance is the same for cumulative logit and probit models but probit model fit better than logit (LR is smaller). We can interpret parameter estimates in terms of the underlying latent variable model. For example, conditional on the Male, the latent distribution on belief is estimated to have location for White that is 0.44936 standard deviations in the belief YES direction compared with that for Black.

- **Q4:** Remained to be done.

Appendix

GWW 12031299

2021/6/9

```
rm(list = ls())
library(glmnet)


y = matrix(ncol=2,nrow=8,c(150,70,60,100,65,134,95,17, 20,30,15,0,55,61,9,33))
X_sat_dummy = rbind(diag(7),rep(0,7)) # 0,1 coding
model_sat_dummy = glm(y~X_sat_dummy, family = "binomial")
summary(model_sat_dummy)
```

```
##
## Call:
## glm(formula = y ~ X_sat_dummy, family = "binomial")
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0  0  0
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     -0.6633     0.2985  -2.222   0.0263 *
## X_sat_dummy1     2.6782     0.3818   7.014 2.31e-12 ***
## X_sat_dummy2     1.5106     0.3698   4.085 4.41e-05 ***
## X_sat_dummy3     2.0496     0.4153   4.935 8.00e-07 ***
## X_sat_dummy4    27.9745 51688.8695   0.001   0.9996
## X_sat_dummy5     0.8303     0.3503   2.371   0.0178 *
## X_sat_dummy6     1.4503     0.3361   4.315 1.60e-05 ***
## X_sat_dummy7     3.0199     0.4591   6.578 4.76e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.6211e+02  on 7  degrees of freedom
## Residual deviance: 2.7540e-10  on 0  degrees of freedom
## AIC: 48.983
##
## Number of Fisher Scoring iterations: 22
```

```
X_sat_effect = rbind(diag(7),rep(-1,7)) # -1,0,1 coding
model_sat_effect = glm(y~X_sat_effect, family = "binomial")
summary(model_sat_effect)
```

```
##
## Call:
```

```
## glm(formula = y ~ X_sat_effect, family = "binomial")
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0  0  0
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)       4.276   6461.111   0.001    0.999
## X_sat_effect1    -2.261   6461.111   0.000    1.000
## X_sat_effect2    -3.429   6461.111  -0.001    1.000
## X_sat_effect3    -2.890   6461.111   0.000    1.000
## X_sat_effect4    23.035  45227.776   0.001    1.000
## X_sat_effect5    -4.109   6461.111  -0.001    0.999
## X_sat_effect6    -3.489   6461.111  -0.001    1.000
## X_sat_effect7    -1.919   6461.111   0.000    1.000
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.6211e+02  on 7  degrees of freedom
## Residual deviance: 2.7536e-10  on 0  degrees of freedom
## AIC: 48.983
##
## Number of Fisher Scoring iterations: 22
```

```
pred_dummy = predict(model_sat_dummy,type = "response",se.fit=T)
model_sat_effect$y # same value
```

```
##         1         2         3         4         5         6         7         8
## 0.8823529 0.7000000 0.8000000 1.0000000 0.5416667 0.6871795 0.9134615 0.3400000
```

```
odds_38_dummy = exp(model_sat_dummy$coefficients[4])
CI_low = exp(model_sat_dummy$coefficients[4]-1.96*0.4153) # beta ± Za/2(SE)
CI_up = exp(model_sat_dummy$coefficients[4]+1.96*0.4153)
print(paste('The CI is: (', round(CI_low,4),',',round(CI_up,4), ')'))
```

```
## [1] "The CI is: ( 3.4404 , 17.5241 )"
```

```
magician = factor(rep(c("Ammar","Blaine","Cyril","Green"),2),levels = c("Ammar","Blaine","Cyril","Green
magic = factor(rep(c("Card","Coin"),c(4,4)),levels =c("Card","Coin"))
reduce_model = glm(y~magician+magic,family = "binomial")
summary(reduce_model)
```

```
##
## Call:
## glm(formula = y ~ magician + magic, family = "binomial")
##
## Deviance Residuals:
##      1       2       3       4       5       6       7       8
##  1.823  -3.359  -3.812   5.718  -1.575   2.025   2.776  -4.326
##
## Coefficients:
##                Estimate Std. Error z value Pr(>|z|)
## (Intercept)     1.60295    0.16774   9.556  < 2e-16 ***
## magicianBlaine  0.02234    0.19340   0.116    0.908
## magicianCyril   1.03975    0.26494   3.924 8.69e-05 ***
## magicianGreen   0.12518    0.24589   0.509    0.611
```

```
## magicCoin      -1.14559    0.17365  -6.597 4.19e-11 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 162.108  on 7  degrees of freedom
## Residual deviance:  94.841  on 3  degrees of freedom
## AIC: 137.82
##
## Number of Fisher Scoring iterations: 5
```

```
pchisq(deviance(reduce_model),df.residual(reduce_model),lower.tail=FALSE) # 1.997114e-20
```

```
## [1] 1.997114e-20
```

```
X2 = sum(rstandard(reduce_model, type="pearson")^2)
pchisq(X2,df.residual(reduce_model),lower.tail=FALSE)
```

```
## [1] 2.808566e-48
```

```
reduce_se = c(0.16774,0.19340,0.26494,0.24589,0.17365)
reduce_CI_low = reduce_model$coefficients - 1.96*reduce_se
reduce_CI_up = reduce_model$coefficients + 1.96*reduce_se
print(paste('The CIs are: (', round(reduce_CI_low,4),',',round(reduce_CI_up,4), ')'))
```

```
## [1] "The CIs are: ( 1.2742 , 1.9317 )"   "The CIs are: ( -0.3567 , 0.4014 )"
## [3] "The CIs are: ( 0.5205 , 1.559 )"    "The CIs are: ( -0.3568 , 0.6071 )"
## [5] "The CIs are: ( -1.4859 , -0.8052 )"
```

```
saturated_model = glm(y~magician + magic + magician:magic,family = "binomial") # model_sat_dummy
summary(saturated_model) # Wald test p-values
```

```
##
## Call:
## glm(formula = y ~ magician + magic + magician:magic, family = "binomial")
##
## Deviance Residuals:
## [1]  0  0  0  0  0  0  0  0
##
## Coefficients:
##                            Estimate Std. Error z value Pr(>|z|)
## (Intercept)                  2.0149     0.2380   8.464  < 2e-16 ***
## magicianBlaine              -1.1676     0.3229  -3.616   0.0003 ***
## magicianCyril               -0.6286     0.3742  -1.680   0.0930 .
## magicianGreen               25.2963 51688.8856   0.000   0.9996
## magicCoin                   -1.8478     0.3004  -6.152 7.67e-10 ***
## magicianBlaine:magicCoin     1.7875     0.4021   4.445 8.78e-06 ***
## magicianCyril:magicCoin      2.8182     0.5433   5.187 2.14e-07 ***
## magicianGreen:magicCoin    -26.1266 51688.8856  -0.001   0.9996
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1.6211e+02  on 7  degrees of freedom
## Residual deviance: 2.7536e-10  on 0  degrees of freedom
```

```
## AIC: 48.983
##
## Number of Fisher Scoring iterations: 22
```

```
pchisq(deviance(reduce_model)-deviance(saturated_model),df=df.residual(reduce_model)-df.residual(satura
```

```
## [1] 1.997114e-20
```

```
library(VGAM)
belief_data = read.table("data/Belief.txt",header=T)
## model selection: find out a baseline model, main effects
saturated = vglm(cbind(Unsure,No,Yes)~ Race+Gender+Race:Gender,family=multinomial,data=belief_data)
model_null = vglm(cbind(Unsure,No,Yes) ~ 1,family=multinomial, data=belief_data)
model_G = vglm(cbind(Unsure,No,Yes)~Gender,family=multinomial, data=belief_data)
model_R = vglm(cbind(Unsure,No,Yes)~ Race,family=multinomial, data=belief_data)
model_GR = vglm(cbind(Unsure,No,Yes) ~ Gender+Race,family=multinomial, data=belief_data)

pchisq(deviance(model_null)-deviance(model_G),df=df.residual(model_null)-df.residual(model_G),lower.tai
```

```
## [1] 4.602426e-10
```

```
pchisq(deviance(model_null)-deviance(model_R),df=df.residual(model_null)-df.residual(model_R),lower.tai
```

```
## [1] 1.120404e-07
```

```
pchisq(deviance(model_G)-deviance(saturated),df=df.residual(model_G)-df.residual(saturated),lower.tail=
```

```
## [1] 3.156011e-07
```

```
pchisq(deviance(model_R)-deviance(saturated),df=df.residual(model_R)-df.residual(saturated),lower.tail=
```

```
## [1] 1.673184e-09
```

```
pchisq(deviance(model_G)-deviance(model_GR),df=df.residual(model_G)-df.residual(model_GR),lower.tail=FA
```

```
## [1] 3.480165e-07
```

```
pchisq(deviance(model_R)-deviance(model_GR),df=df.residual(model_R)-df.residual(model_GR),lower.tail=FA
```

```
## [1] 1.429592e-09
```

```
pchisq(deviance(model_GR),df.residual(model_GR),lower.tail=FALSE) # 0.04796053
```

```
## [1] 0.04796053
```

```
sqrt(deviance(model_GR)/df.residual(model_GR))
```

```
## [1] 1.742807
```

```
model_nomial = model_GR
summary(model_nomial, cor=T)
```

```
##
## Call:
## vglm(formula = cbind(Unsure, No, Yes) ~ Gender + Race, family = multinomial,
##     data = belief_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1 -0.26345    0.09589  -2.747  0.00601 **
## (Intercept):2 -1.79431    0.16751 -10.712  < 2e-16 ***
```

```
## Gender:1      -0.72521    0.13183  -5.501 3.77e-08 ***
## Gender:2      -1.03390    0.25866  -3.997 6.41e-05 ***
## Race:1        -1.14842    0.23675  -4.851 1.23e-06 ***
## Race:2        -0.67270    0.41144  -1.635  0.10205
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])
##
## Residual deviance: 6.0748 on 2 degrees of freedom
##
## Log-likelihood: -21.7917 on 2 degrees of freedom
##
## Number of Fisher scoring iterations: 4
##
## Correlation of Coefficients:
##
##               (Intercept):1 (Intercept):2 Gender:1 Gender:2 Race:1
## (Intercept):2  0.2424
## Gender:1      -0.6952       -0.1666
## Gender:2      -0.1459       -0.6036        0.1695
## Race:1        -0.2106       -0.0470        0.0023   0.0038
## Race:2        -0.0531       -0.2501        0.0126  -0.0131   0.1019
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Reference group is level  3  of the response
```

```
summary(model_nomial,dispersion=1.74) # ? useful
```

```
##
## Call:
## vglm(formula = cbind(Unsure, No, Yes) ~ Gender + Race, family = multinomial,
##     data = belief_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  -0.2634     0.1265  -2.083 0.037270 *
## (Intercept):2  -1.7943     0.2210  -8.121 4.64e-16 ***
## Gender:1       -0.7252     0.1739  -4.170 3.04e-05 ***
## Gender:2       -1.0339     0.3412  -3.030 0.002444 **
## Race:1         -1.1484     0.3123  -3.677 0.000236 ***
## Race:2         -0.6727     0.5427  -1.239 0.215169
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: log(mu[,1]/mu[,3]), log(mu[,2]/mu[,3])
##
## Dispersion Parameter for multinomial family:   1.74
##
## Residual deviance: 6.0748 on 2 degrees of freedom
##
## Log-likelihood: -21.7917 on 2 degrees of freedom
##
```

```
## Number of Fisher scoring iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Reference group is level  3  of the response
```

```
resid(model_nomial,type="response") # the difference between the observed values and the fitted values
```

```
##         Unsure           No          Yes
## 1  0.048171803 -0.007408629 -0.040763175
## 2 -0.077366835  0.011898706  0.065468129
## 3 -0.009085785  0.001397357  0.007688428
## 4  0.011028534 -0.001696144 -0.009332390
```

```
resid(model_nomial,type="stdres") # extracts the table of count, then (observed -expected) / sqrt(V), V
```

```
##      Unsure         No       Yes
## 1 -3.404604 -1.8205109  4.141407
## 2 -3.461186  0.6154113  2.999650
## 3 -3.141646 -2.1881562  4.071030
## 4  6.827601  3.0158715 -7.993488
```

```
predict(model_nomial,type="response")
```

```
##      Unsure         No       Yes
## 1 0.1027716 0.02627655 0.8709519
## 2 0.1834274 0.06385887 0.7527137
## 3 0.2599755 0.04130727 0.6987173
## 4 0.3971788 0.08592941 0.5168918
```

```
## interpret the conditional gender and race effects respectively
t(coef(model_nomial,matrix=T))
```

```
##                     (Intercept)    Gender      Race
## log(mu[,1]/mu[,3])   -0.263447 -0.725212 -1.148419
## log(mu[,2]/mu[,3])   -1.794307 -1.033900 -0.672702
```

```
rev(cumsum(rev(coef(model_nomial,matrix=T)["(Intercept)",]))) # These are the alpha_j for the baseline-
```

```
## log(mu[,1]/mu[,3]) log(mu[,2]/mu[,3])
##          -2.057754          -1.794307
```

```
rev(cumsum(rev(coef(model_nomial,matrix=T)["Gender",]))) # effects for Gender
```

```
## log(mu[,1]/mu[,3]) log(mu[,2]/mu[,3])
##          -1.759112          -1.033900
```

```
rev(cumsum(rev(coef(model_nomial,matrix=T)["Race",]))) # effects for Race
```

```
## log(mu[,1]/mu[,3]) log(mu[,2]/mu[,3])
##          -1.821121          -0.672702
```

```
model_logit = vglm(cbind(Yes,Unsure,No)~ Race+Gender,family=cumulative(parallel=T),data=belief_data)
pchisq(deviance(model_logit),df.residual(model_logit),lower.tail=FALSE) # 0.05505123
```

```
## [1] 0.05505123
```

```
summary(model_logit)
```

```
##
```

```
## Call:
## vglm(formula = cbind(Yes, Unsure, No) ~ Race + Gender, family = cumulative(parallel = T),
##     data = belief_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1  0.07631    0.08963   0.851    0.395
## (Intercept):2  2.32238    0.13522  17.175  < 2e-16 ***
## Race           1.01645    0.21059   4.827 1.39e-06 ***
## Gender         0.76956    0.12253   6.281 3.37e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: logitlink(P[Y<=1]), logitlink(P[Y<=2])
##
## Residual deviance: 9.2542 on 4 degrees of freedom
##
## Log-likelihood: -23.3814 on 4 degrees of freedom
##
## Number of Fisher scoring iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
##     Race   Gender
## 2.763372 2.158827
```

```
model_probit = vglm(cbind(Yes,Unsure,No)~ Race+Gender,family=cumulative(link="probitlink",reverse=T,para
pchisq(deviance(model_probit),df.residual(model_probit),lower.tail=FALSE) # 0.01071517
```

```
## [1] 0.01071517
```

```
summary(model_probit)
```

```
##
## Call:
## vglm(formula = cbind(Yes, Unsure, No) ~ Race + Gender, family = cumulative(link = "probitlink",
##     reverse = T, parallel = T), data = belief_data)
##
## Coefficients:
##               Estimate Std. Error z value Pr(>|z|)
## (Intercept):1 -0.06776    0.05486  -1.235    0.217
## (Intercept):2 -1.30293    0.06828 -19.081  < 2e-16 ***
## Race          -0.54371    0.11520  -4.720 2.36e-06 ***
## Gender        -0.44936    0.07201  -6.241 4.36e-10 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Names of linear predictors: probitlink(P[Y>=2]), probitlink(P[Y>=3])
##
## Residual deviance: 13.1176 on 4 degrees of freedom
##
## Log-likelihood: -25.3131 on 4 degrees of freedom
##
```

```
## Number of Fisher scoring iterations: 4
##
## No Hauck-Donner effect found in any of the estimates
##
##
## Exponentiated coefficients:
##      Race    Gender
## 0.5805920 0.6380384
```