Ordinal data and a comparison of the General Estimating Equations against Maximum Likelihood Estimation

Sam Tracy

Abstract

In this paper we consider analyzing ordinal data using the Proportional Odds model. Our interest is in the efficiency of the General Estimating Equations (GEE) method versus Maximum Likelihood Estimation (MLE). Using an appropriate simulation without repeated measures we show that the GEE results follow those of the MLE quite closely. We conclude that the difference in efficiency is negligible when not using repeated measures, and place further interest in the efficiency of the GEE method otherwise.

Acknowledgements

My advisor, Professor Ronghui Xu, for her countless hours counseling me through the last two years of my education, as well as this thesis. Both my future and current interests have been made possible by her guidance (and patience). Professor Adrian Ioana, for inspiring me with an appreciation for analysis. My old roommate, Mike Waterson, for encouraging me to attempt any of this in the first place.

Contents

1 Introduction

The use of ordinal data is prevalent in the public health fields. In a recent study at the CTRI[1], data was collected from women with pelvic floor disorders before surgery, and at six and twelve weeks after surgery. A portion of the data was collected via a series of questionnaires, and the researchers were interested in analyzing a specific set of responses on the questionnaires, which had an ordinal score of 0, 1, 2, 3, or 4. The analysis used the Generalized Estimating Equations (GEE) approach, given the repeated measures. The statisticians found the R function *repolr*{*repolr*} [1] to fit the repeated measures ordinal data, which converts the ordinal data into binary pseudo-data before applying the GEE approach. The purpose of this paper is to examine this method in a simpler setting of i.i.d. data without repeated measures in order to compare its efficiency to the asymptotically efficient Maximum Likelihood Estimation (MLE). Given the theoretical optimality of the MLE, we expect that the GEE will follow in performance yet are interested in measuring the difference by which it does. The result may provide some indication as to how they might compare over repeated measures.

2 Theoretical basis

Consider a record of ordinal scores obtained in a clinical study with n subjects assumed to be i.i.d. and K ordered categories. Let Y_i be the response for the i-th individual. Let X_i be a vector of length L containing the observed covariates, specific to individual i. Let β be a corresponding vector of coefficients, and ζ a vector of intercepts between response levels. Denote $p_{ik} = P(Y_i = k|X_i = x)$. We define the odds of k to be the ratio $p_{ik}/(1 - p_{ik})$. The proportional odds model with $K = 2$ and $L = 1$ has:

$$P(Y_i = k|X_i = x) = \frac{\exp(\zeta_k + \beta x)}{1 + \exp(\zeta_k + \beta x)} \ , \quad k = 1, 2 \tag{1}$$

where $-\infty < \beta < \infty$. Taking the logit[2] of the above ratio as a link function, or transformation, we obtain:

$$\text{logit}\{P(Y_i = k|X_i = x)\} = \zeta_k + \beta x \ , \tag{2}$$

allowing us to view the components through a generalized linear model. Now, for $K \geq 2$, we may derive odds ratios in a number of ways. One might consider the proportion between multinomial, cumulative, or adjacent probabilities. It is

[1] Clinical and Translational Research Institute, UC San Diego

[2] $\text{logit}(p) = \log\{p/(1 - p)\}$

worth noting that each of these derivations utilize the same basic structure, but the values for the ζ intercepts may vary between the three possibilities. We will use the cumulative odds. In this particular case, we note that as the cumulative probabilities must increase, ζ_k must strictly increase with k. We consider $P(Y_i \leq k)$ and obtain the following cumulative logit:

$$\operatorname{logit}\{P(Y_i \leq k|X_i)\} = \zeta_k + \beta_1 x_{i1} + \beta_2 x_{i2} + ...\beta_L x_{iL} \tag{3}$$

We note the conceptual basis behind the proportional odds model by relating it to the latent variable Z, having a logistic distribution with $P(Y \leq k) = P(Z \leq z_k)$. Then:

$$P(Y \leq k) = P(Z \leq z_k) = \frac{\exp{(z_k)}}{1+\exp{(z_k)}} = \frac{\exp{(\zeta_k + \beta' x)}}{1+\exp{(\zeta_k + \beta' x)}} \tag{4}$$

We may interpret β as the proportional odds between groups, and the ζ as 'baseline' odds. An assumption of the model is that our covariate coefficients are constant across each level, while ζ_k varies with k, indicating the intercept for proportional odds between response levels. In a simplistic setting with $K = 2$ and $L = 1$ this may be interpreted as the relationship between predictor and binary response, treatment and outcome. Extending this model to $K \geq 2$ allows us to consider more refined structures of treatment and outcome, such as scales of quality. We may also extend L and consider factors such as age[3] and gender.

2.1 Maximum Likelihood Estimation

Allowing $F_{ik} = \{P(Y_i \leq k|X_i)\}$, we have $p_{ik} = F_{i,k} - F_{i,k-1}$ and the likelihood function is:

$$\prod_{i=1}^{n}\prod_{k=1}^{K}\{p_{ik}\,(\zeta_k,\beta)\}^{y_{ik}} \tag{5}$$

The ζ intercepts and β coefficients are then estimated by maximizing this function.

2.2 General Estimating Equations

The GEE is designed for correlated data and does not assume independence. We consider the previous Y_i, with $\operatorname{logit}\{P(Y_i \leq k|X_i)\} = \zeta_k + \beta' X_i$, and use it to create a vector of binary variables:

$$Y_i^* = \begin{pmatrix} Y_{i1}^* \\ \vdots \\ Y_{i(K-1)}^* \end{pmatrix} \quad \text{where} \quad Y_{ik}^* = \begin{cases} 1 & : Y_i > k \\ 0 & : Y_i \leq k \end{cases} . \tag{6}$$

[3] Covariates may be discrete, continuous, or neither

Let $g^{-1}(\mu_{ik}) = \mathrm{E}[Y_{ik}^{*}]$, the cumulative link function relative to our underlying model. Let V_i be the covariance structure for the predictors. Then we have:

$$\begin{aligned} \sum_i \frac{\partial g(\mu_{ik})}{\partial(\beta)} V_i^{-1}\{Y_{ik}^{*} - g^{-1}(\mu_{ik})\} = 0 \\ \sum_i \frac{\partial g(\mu_{ik})}{\partial(\zeta)} V_i^{-1}\{Y_{ik}^{*} - g^{-1}(\mu_{ik})\} = 0 \end{aligned} \tag{7}$$

for each $k = 1, 2, ..., (K-1)$. β and ζ may now be estimated by these systems of equations using an iterative Newton-Raphson method. As the covariance structure is generally unknown, it is treated as a nuisance parameter and the resulting GEE estimations are consistent regardless of the specified structure. However, if the covariance structure is correctly specified, the method may return a useful estimate of the data correlation. This is favorable in the setting of repeated measures.

3 Implementation in R

Having outlined our underlying models and methods, we consider their implementation in the software package R.

3.1 polr()

polr$\{MASS\}$ [2] performs logistic regression using the proportional odds model (3) and the likelihood function (5). There is a slight methodical alteration in using the parameter $\eta = -\beta$ and so we adjust our code accordingly for simulation. Employing the MLE method, *polr* estimates are asymptotically normally distributed and efficient.

3.2 gee()

gee$\{gee\}$ [3] extends the Generalized Linear Model to apply the Generalized Estimating Equations strategy outlined in Section 2.2. As implied, we manipulate the ordinal data into $K-1$ binary pseudo data points that are clustered. Each response is paired with a vector of binary variables using equation (6). This creates a new response vector of length $n(K-1)$. We also number the response clusters and create a design vector to distinguish between levels for each response. The cumulative logit link and binomial family are passed into the function to identify the assumed underlying model and respective transformation for the use of equations (7). This alteration forces applicability of the *gee* in regards to ordinal data. However, it is worth noting that the *gee* package does not calculate multiple intercept estimates. The function only produces ζ_{k-1} and a coefficient of the design vector. Thus, a few minor calculations are necessary to extract the remaining estimates.

4 Simulation

For simulation we consider $K = 3, L = 2$. Two Bernoulli covariates, x_1 and x_2, are generated using *sample* with the respective probabilities of 0.49 and 0.51. Appropriate values for the β coefficients and ζ intercepts are also selected such that the resulting response samples will include each level. Simulating samples in R, we selected $\beta = \{-2.05, 2.05\}$ and $\zeta = \{-0.7, 0.7\}$. The proportional odds model becomes:

$$\text{logit}\{P(Y_i \leq k|X_i)\} = \zeta_k + \beta_1 x_{i1} + \beta_2 x_{i2} \qquad \begin{array}{l} k = 1, 2, 3 \\ i = 1, 2, ..., n \end{array} \tag{8}$$

Using (8) as the true model, we create probability vectors for the multinomial probabilities $p_{ik} = P(Y_i = k) = P(Y_i \leq k) - P(Y_i \leq k-1)$, for each $1 \leq k \leq K$:

$$\begin{aligned} p_{i1} &= \frac{\exp(\zeta_1 + \beta X_i)}{1 + \exp(\zeta_1 + \beta X_i)} \\ p_{i2} &= \frac{1}{1 + \exp(\zeta_1 + \beta X_i)} - \frac{1}{1 + \exp(\zeta_2 + \beta X_i)} \\ p_{i3} &= 1 - p_{i1} - p_{i2} \end{aligned} \tag{9}$$

We then use these probabilities to generate a response vector with $K = 3$ levels and probabilities corresponding to each level given the covariate values. The *polr* function is applied to the data and the MLE estimates are stored. The true data are then transformed into 'new' correlated binary data, as discussed in Section 3.2, and *gee* applied. After all the estimates are stored, the process is repeated N times and the results tabulated. For this paper we consider sample sizes of $n = 200, 500, 1000, 2000$ at $N = 5000$. The R code used in this simulation is included in the appendix. Comparing the results of the *gee* and *polr* functions across varying sample sizes in Tables 1 - 4 we see that, while the GEE method is consistently less accurate than MLE as we had expected, it does indeed approach the MLE. The standard deviation (SD), standard error (SE), and mean-squared error (MSE) of each $\hat{\zeta}_k$ and $\hat{\beta}$ are quite close to those of the MLE, with coverage probabilities of 95% confidence intervals (CP) approaching 0.95. Comparing the estimate for ζ_1 in Tables 1 and 4, we see that the respective SD, SE, and MSE of {0.255, 0.256, 0.065} at $n = 200$ are reduced to {0.082, 0.080, 0.007} at $n = 2000$ under the *polr* function, while the corresponding values {0.255, 0.258, 0.065} at $n = 200$ are reduced to {0.082, 0.080, 0.007} at $n = 2000$ using the *gee* function.

5 Discussion

In this paper we have outlined the theory and methods for MLE and the GEE regarding ordinal scores and assuming the proportional odds model, briefly discussing the benefits and complications of their usage. We observe that the GEE closely follows the MLE in efficiency during simulation, with the difference between estimates becoming virtually indistinguishable for large sample sizes. From these results we conclude that the GEE is relatively efficient compared to MLE in estimating ζ_k and β with ordinal data, placing further interest in how efficient the GEE is over repeated measures.

References

[1] Parsons, N. R., Costa, M. L., Achten, J. and Stallard, N. (2009) Repeated measures proportional odds logistic regression analysis of ordinal score data in the statistical software package R. *Computational Statistics & Data Analysis*, Vol.53 (No.3). pp. 632-641. ISSN 0167-9473

[2] Venables, W. N. and Ripley, B. D. (2002). *Modern Applied Statistics with S.* Fourth Edition. Springer, New York. ISBN 0-387-95457-0

[3] Carey, V. J. Ported to R by Thomas Lumley and Brian Ripley. (2012). *gee: Generalized Estimation Equation solver. R package version 4.13-18.* http://CRAN.R-project.org/package=gee

DEPARTMENT OF MATHEMATICS
UNIVERSITY OF CALIFORNIA, SAN DIEGO
E-mail address: stracy@ucsd.edu

Appendix

```
library(MASS)
library(gee)

N=5000
n=2000
#Zeta and Beta coefficients
#Note : Zeta for logit*P( Y <= k )
Z=c(-0.7, 0.7)
B=c(2.05, -2.05)

#result data frame
dfp <- as.data.frame(matrix(nrow=8, ncol=6))

#result vectors
X1 <- numeric(N)
Y1 <- numeric(N)
U1 <- numeric(N)
V1 <- numeric(N)
X2 <- numeric(N)
Y2 <- numeric(N)
U2 <- numeric(N)
V2 <- numeric(N)

#CI vectors
CIX1 <- numeric(N)
CIY1 <- numeric(N)
CIU1 <- numeric(N)
CIV1 <- numeric(N)
CIX2 <- numeric(N)
CIY2 <- numeric(N)
CIU2 <- numeric(N)
CIV2 <- numeric(N)

#error vectors
eX1 <- numeric(N)
eY1 <- numeric(N)
eU1 <- numeric(N)
eV1 <- numeric(N)
eX2 <- numeric(N)
eY2 <- numeric(N)
eU2 <- numeric(N)
eV2 <- numeric(N)

#probability vectors
p1 <- numeric(n)
p2 <- numeric(n)
p3 <- numeric(n)

#response vectors
response <- numeric(n)
y1 <- numeric(n)
y2 <- numeric(n)
res <- numeric(2*n)

#design vectors
identity <- numeric(2*n)
des <- numeric(2*n)
tx2 <- numeric(2*n)
sex2 <- numeric(2*n)

for (iter in 1:N) {
#generate treatment and gender covariates
```

```
tx <- sample(c(0,1), n, replace=TRUE, prob = c(.49,.51))
sex <- sample(c(0,1), n, replace=TRUE, prob = c(.51,.49))
#construct probabilities according to logit model
for (i in 1:n){
p1[i] = exp(Z[1]-(B[1]*tx[i]+B[2]*sex[i]))/(1+exp(Z[1]-(B[1]*tx[i]+B[2]*sex[i])))
p2[i] = 1/(1+exp(Z[1]-B[1]*tx[i]-B[2]*sex[i]))-1/(1+exp(Z[2]-B[1]*tx[i]-B[2]*sex[i]))
p3[i] = (1-p1[i]-p2[i])
}

#build ordinal responses from probabilities
for (i in 1:n){
response[i] <- sample( 1:3, 1, replace=TRUE, c(p1[i],p2[i],p3[i]) )
}

#fit data to polr and add to result matrices
response = as.factor(response)
pfit <- polr(response ~ tx + sex, Hess=TRUE, model=TRUE, method="logistic")

#store estimate data
X1[iter] = as.double(pfit$zeta[1])
X2[iter] = as.double(pfit$zeta[2])
Y1[iter] = as.double(pfit$coef[1])
Y2[iter] = as.double(pfit$coef[2])
#std. error
eX1[iter] <- as.double(summary(pfit)$coefficients[3,2])
eX2[iter] <- as.double(summary(pfit)$coefficients[4,2])
eY1[iter] <- as.double(summary(pfit)$coefficients[1,2])
eY2[iter] <- as.double(summary(pfit)$coefficients[2,2])

#construct confidence interval and store result
if( Z[1] < X1[iter]+1.96*eX1[iter] && Z[1] > X1[iter]-1.96*eX1[iter] )
{ CIX1[iter]=1 }
if( Z[2] < X2[iter]+1.96*eX2[iter] && Z[2] > X2[iter]-1.96*eX2[iter] )
{ CIX2[iter]=1 }
if( B[1] < Y1[iter]+1.96*eY1[iter] && B[1] > Y1[iter]-1.96*eY1[iter] )
{ CIY1[iter]=1 }
if( B[2] < Y2[iter]+1.96*eY2[iter] && B[2] > Y2[iter]-1.96*eY2[iter] )
{ CIY2[iter]=1 }

#convert ordinal data to a series of binary vectors
#construct new response and covariate vectors accordingly
j=1
for(k in 1:n) {
if(response[k]==1) {y1[k]=0; y2[k]=0}
if(response[k]==2) {y1[k]=0; y2[k]=1}
if(response[k]==3) {y1[k]=1; y2[k]=1}
res[2*k-1]=y1[k]
res[2*k]=y2[k]
identity[2*k]=j
identity[2*k-1]=j
j=j+1

tx2[2*k-1]=tx[k]
tx2[2*k]=tx[k]
sex2[2*k-1]=sex[k]
sex2[2*k]=sex[k]
des[2*k]=1
des[2*k-1]=0
}

#fit new psuedo data to GEE
gfit <- gee(res ~ des + tx2 + sex2, id=identity, family = "binomial",
corstr="exchangeable", silent=TRUE)

#store estimate data
```

```
#retrieving intercepts from given GEE value
U1[iter] = as.double(-gfit$coef[1]-gfit$coef[2])
U2[iter] = as.double(-gfit$coef[1])
V1[iter] = as.double(gfit$coef[3])
V2[iter] = as.double(gfit$coef[4])

#calculate standard error for retrieved intercepts
a = gfit$naive.variance[1,1]
b = gfit$naive.variance[2,2]
c = gfit$naive.variance[1,2]
nst = sqrt(a+b+2*c)

#store std. error
eU1[iter] <- as.double(nst)
eU2[iter] <- as.double(summary(gfit)$coefficients[1,2])
eV1[iter] <- as.double(summary(gfit)$coefficients[3,2])
eV2[iter] <- as.double(summary(gfit)$coefficients[4,2])

#construct confidence interval and store result
if( Z[1] < U1[iter]+1.96*eU1[iter] && Z[1] > U1[iter]-1.96*eU1[iter])
{ CIU1[iter]=1 }
if( Z[2] < U2[iter]+1.96*eU2[iter] && Z[2] > U2[iter]-1.96*eU2[iter])
{ CIU2[iter]=1 }
if( B[1] < V1[iter]+1.96*eV1[iter] && B[1] > V1[iter]-1.96*eV1[iter])
{ CIV1[iter]=1 }
if( B[2] < V2[iter]+1.96*eV2[iter] && B[2] > V2[iter]-1.96*eV2[iter])
{ CIV2[iter]=1 }
}

colnames(dfp) <- c("True", "Estimate", "SD", "SE", "MSE", "CP")
row.names(dfp) <- c("1|2 POLR","1|2 (GEE)","2|3 POLR","2|3 (GEE)",
"Beta 0 (POLR)","Beta 0 (GEE)","Beta 1 (POLR)","Beta 1 (GEE)")

dfp[1,] <- c(Z[1], mean(X1), sd(X1), mean(eX1),
(Z[1]-mean(X1))^2+sd(X1)^2, mean(CIX1))
dfp[3,] <- c(Z[2], mean(X2), sd(X2), mean(eX2),
(Z[2]-mean(X2))^2+sd(X2)^2, mean(CIX2))
dfp[5,] <- c(B[1], mean(Y1), sd(Y1), mean(eY1),
(B[1]-mean(Y1))^2+sd(Y1)^2, mean(CIY1))
dfp[7,] <- c(B[2], mean(Y2), sd(Y2), mean(eY2),
(B[2]-mean(Y2))^2+sd(Y2)^2, mean(CIY2))
dfp[2,] <- c(Z[1], mean(U1), sd(U1), mean(eU1),
(Z[1]-mean(U1))^2+sd(U1)^2, mean(CIU1))
dfp[4,] <- c(Z[2], mean(U2), sd(U2), mean(eU2),
(Z[2]-mean(U2))^2+sd(U2)^2, mean(CIU2))
dfp[6,] <- c(B[1], mean(V1), sd(V1), mean(eV1),
(B[1]-mean(V1))^2+sd(V1)^2, mean(CIV1))
dfp[8,] <- c(B[2], mean(V2), sd(V2), mean(eV2),
(B[2]-mean(V2))^2+sd(V2)^2, mean(CIV2))

dfp
```

Tables

Values at $N = 5000$ simulations

Table 1: $n = 200$

		True	Estimate	SD	SE	MSE	CP
ζ_1	*polr*	-0.70	-0.712	0.255	0.256	0.065	0.952
	gee	-0.70	-0.712	0.255	0.258	0.065	0.954
ζ_2	*polr*	0.70	0.709	0.256	0.256	0.066	0.953
	gee	0.70	0.710	0.256	0.258	0.066	0.953
β_1	*polr*	2.05	2.085	0.329	0.323	0.109	0.949
	gee	2.05	2.092	0.332	0.326	0.112	0.949
β_2	*polr*	-2.05	-2.085	0.330	0.323	0.110	0.948
	gee	-2.05	-2.092	0.333	0.326	0.112	0.949

Table 2: $n = 500$

		True	Estimate	SD	SE	MSE	CP
ζ_1	*polr*	-0.70	-0.703	0.162	0.161	0.026	0.948
	gee	-0.70	-0.703	0.162	0.161	0.026	0.947
ζ_2	*polr*	0.70	0.705	0.163	0.161	0.027	0.946
	gee	0.70	0.705	0.163	0.161	0.027	0.947
β_1	*polr*	2.05	2.06	0.203	0.202	0.041	0.949
	gee	2.05	2.07	0.203	0.203	0.042	0.950
β_2	*polr*	-2.05	-2.065	0.198	0.202	0.040	0.956
	gee	-2.05	-2.067	0.199	0.203	0.040	0.956

Table 3: $n = 1000$

		True	Estimate	SD	SE	MSE	CP
ζ_1	*polr*	-0.70	-0.701	0.116	0.113	0.013	0.948
	gee	-0.70	-0.701	0.116	0.114	0.013	0.948
ζ_2	*polr*	0.70	0.701	0.116	0.113	0.013	0.949
	gee	0.70	0.701	0.116	0.114	0.013	0.950
β_1	*polr*	2.05	2.056	0.143	0.142	0.021	0.948
	gee	2.05	2.057	0.144	0.143	0.021	0.949
β_2	*polr*	-2.05	-2.059	0.142	0.142	0.020	0.953
	gee	-2.05	-2.061	0.143	0.143	0.020	0.954

Table 4: $n = 2000$

		True	Estimate	SD	SE	MSE	CP
ζ_1	*polr*	-0.70	-0.701	0.082	0.080	0.007	0.945
	gee	-0.70	-0.701	0.082	0.080	0.007	0.944
ζ_2	*polr*	0.70	0.701	0.080	0.080	0.006	0.950
	gee	0.70	0.701	0.080	0.080	0.006	0.950
β_1	*polr*	2.05	2.054	0.102	0.100	0.010	0.948
	gee	2.05	2.055	0.102	0.101	0.010	0.947
β_2	*polr*	-2.05	-2.055	0.101	0.100	0.010	0.949
	gee	-2.05	-2.056	0.101	0.101	0.010	0.950