Day 9 - Extensions

David Carlson

December 3, 2020

Quasi-Likelihood Estimation

- There are cases when there is not enough information about the distribution of the data, or the parametric form of the likelihood is known to be misspecified
- Precludes the standard maximum likelihood estimation of unknown parameters since we cannot specify a full likelihood equation or a score function
- Quasi-likelihood only requires specification of the mean function of the data and a stipulated relationship between this mean function and the variance function
- Quasi-score function: $q_i = \frac{y_i - \mu_i}{a(\psi)\tau^2}$
- Contribution of ith point to log-likelihood function: $Q_i = \int_{y_i}^{\mu_i} \frac{y_i - \mu_i}{a(\psi)\tau^2} dt$
- Components of $\mathbf{Y}$ are independent by assumption (we can violate this in later weeks), the log-quasi-likelihood for the complete data is the sum of the individual contributions: $Q(\theta, a(\psi)|y) = \sum_{i=1}^{n} Q_i$
- MLE of $\hat{\theta}$: $\frac{\partial}{\partial\theta} Q(\theta, \psi|y) = -\sum_{i=1}^{n} y_i + n\theta \equiv 0$
- Quasi-deviance function: $D(\theta, \psi|y) = -2a(\psi)^{-1} \sum_{i=1}^{n} Q_i = 2\int_{\mu_i}^{y_i} \frac{y_i^{-t}}{\tau^2} dt$
- Table 7.1 has some common quasi-likelihoods
- Quasi-likelihood estimator is often less efficient than MLE and can never be more efficient
- Quasi-Poisson: When there is overdispersion, allows us to model the variance as a linear function of the mean in contrast to the underlying assumption of a Poisson model that $\mu = \tau^2$ (can account for outliers)

```
pop_logit = glm(intercon ~ country_pop + aggdifxx + gdppc + polity2,
                data=data,
                family = binomial)
summary(pop_logit) #notice (Dispersion parameter for binomial family taken to be 1)
```

```
##
## Call:
## glm(formula = intercon ~ country_pop + aggdifxx + gdppc + polity2,
##     family = binomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7248  -1.2371   0.9087   1.0745   1.3535
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.071172   0.058710  -1.212    0.225
## country_pop  0.451093   0.038043  11.857  < 2e-16 ***
## aggdifxx     0.037667   0.005560   6.775 1.25e-11 ***
## gdppc       -0.141813   0.035302  -4.017 5.89e-05 ***
## polity2      0.017915   0.003707   4.833 1.35e-06 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11183  on 8215  degrees of freedom
## Residual deviance: 10786  on 8211  degrees of freedom
##   (8652 observations deleted due to missingness)
## AIC: 10796
##
## Number of Fisher Scoring iterations: 5
```

```
q_pop_logit = glm(intercon ~ country_pop + aggdifxx + gdppc + polity2,
                  data=data,
                  family = quasibinomial)
cbind(coef(pop_logit), coef(q_pop_logit)) #equivalent
```

```
##                    [,1]        [,2]
## (Intercept) -0.07117167 -0.07117167
## country_pop  0.45109331  0.45109331
## aggdifxx     0.03766738  0.03766738
## gdppc       -0.14181284 -0.14181284
## polity2      0.01791514  0.01791514
```

```
cbind(confint(pop_logit), confint(q_pop_logit)) #suggests there is no problem
```

```
## Waiting for profiling to be done...
## Waiting for profiling to be done...

##                   2.5 %      97.5 %       2.5 %      97.5 %
## (Intercept) -0.18628399  0.04387801 -0.18789015  0.04548242
## country_pop  0.37922345  0.52852783  0.37825776  0.52964952
## aggdifxx     0.02678184  0.04857908  0.02663018  0.04873147
## gdppc       -0.21104890 -0.07257936 -0.21201575 -0.07161254
## polity2      0.01065366  0.02518571  0.01055244  0.02528718
```

```
summary(q_pop_logit) #notice (Dispersion parameter for quasibinomial family taken to be 1.028087), so n
```

```
##
## Call:
## glm(formula = intercon ~ country_pop + aggdifxx + gdppc + polity2,
##     family = quasibinomial, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7248  -1.2371   0.9087   1.0745   1.3535
##
## Coefficients:
##              Estimate Std. Error t value Pr(>|t|)
## (Intercept) -0.071172   0.059529  -1.196    0.232
## country_pop  0.451093   0.038573  11.694  < 2e-16 ***
## aggdifxx     0.037667   0.005638   6.681 2.52e-11 ***
## gdppc       -0.141813   0.035794  -3.962 7.50e-05 ***
## polity2      0.017915   0.003759   4.766 1.91e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasibinomial family taken to be 1.028087)
```

```
##
##     Null deviance: 11183  on 8215  degrees of freedom
## Residual deviance: 10786  on 8211  degrees of freedom
##   (8652 observations deleted due to missingness)
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```r
#let's pretend polity2 is a count, and use gdppc to model it using a Poisson
pop_pois = glm(I(polity2 + 10) ~ gdppc, data = data,
               family = poisson)
summary(pop_pois) #(Dispersion parameter for poisson family taken to be 1)
```

```
##
## Call:
## glm(formula = I(polity2 + 10) ~ gdppc, family = poisson, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.1583  -2.4657   0.1236   1.8664   2.9523
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 2.340988   0.002941  795.86   <2e-16 ***
## gdppc       0.213665   0.002447   87.33   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for poisson family taken to be 1)
##
##     Null deviance: 64742  on 11239  degrees of freedom
## Residual deviance: 58546  on 11238  degrees of freedom
##   (5628 observations deleted due to missingness)
## AIC: 101271
##
## Number of Fisher Scoring iterations: 5
```

```r
q_pop_pois = glm(I(polity2 + 10) ~ gdppc, data = data,
               family = quasipoisson)
summary(q_pop_pois) #(Dispersion parameter for quasipoisson family taken to be 4.470922) != 1
```

```
##
## Call:
## glm(formula = I(polity2 + 10) ~ gdppc, family = quasipoisson,
##     data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -7.1583  -2.4657   0.1236   1.8664   2.9523
##
## Coefficients:
##             Estimate Std. Error t value Pr(>|t|)
## (Intercept) 2.340988   0.006220   376.4   <2e-16 ***
## gdppc       0.213665   0.005173    41.3   <2e-16 ***
## ---
```

```
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for quasipoisson family taken to be 4.470922)
##
##     Null deviance: 64742  on 11239  degrees of freedom
## Residual deviance: 58546  on 11238  degrees of freedom
##   (5628 observations deleted due to missingness)
## AIC: NA
##
## Number of Fisher Scoring iterations: 5
```

```
cbind(confint(pop_pois), confint(q_pop_pois)) #larger CIs
```

```
## Waiting for profiling to be done...
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %     2.5 %    97.5 %
## (Intercept) 2.3352174 2.3467477 2.3287729 2.3531533
## gdppc       0.2088577 0.2184481 0.2034722 0.2237511
```

```
#?family

#let's compare to negative binomial (for overdispersed counts)
library(MASS)
pop_nb = glm.nb(I(polity2 + 10) ~ gdppc, data = data)
cbind(coef(q_pop_pois), coef(pop_nb))
```

```
##                  [,1]      [,2]
## (Intercept) 2.3409882 2.3380395
## gdppc       0.2136648 0.2669946
```

```
cbind(confint(q_pop_pois), confint(pop_nb))
```

```
## Waiting for profiling to be done...
## Waiting for profiling to be done...
```

```
##                 2.5 %    97.5 %     2.5 %    97.5 %
## (Intercept) 2.3287729 2.3531533 2.3236458 2.3524803
## gdppc       0.2034722 0.2237511 0.2487714 0.2854645
```

```
#Since the negative binomial distribution has one more parameter than the Poisson, the second parameter
# In the case of modest overdispersion, this may produce substantially similar results to an overdispers
```

Generalized Linear Mixed-Effects Model

- We will deal with in more detail in the TSCS week and in Bayesian
- Mixed-effects models consider the dependencies of the observations within clusters and allow us not only to reach unbiased estimates of the effect of covariates of interest and their respective standard errors but also to address questions related to the variation between and within groups: analyze the trajectories of groups/individuals through time, assess the differences between clusters, and others
- This approach is useful for panel data where responses recorded through time are perfectly grouped by panelist
- For GLMMs, we add random effects to the linear predictor and then express the expected value of the outcome conditional on those random effects
  - Effect of being a unit of observation (if the random effect is at the unit-level)

- If the subjects in our sample have been chosen randomly with the goal of treating them as a representation of the population of interest, then their effects on the outcome are also going to be random and generalizable to that same population
  - Random variable that not only will help to make inferences about the population but also allows us to assess the variation between individuals, predict outcomes for each of them, and incorporate the existent correlation between observations
  - There is therefore a distributional assumption on random effects (as opposed to with fixed effects)
  - Generally more power than fixed effects, but need to make the above assumptions, because with greater power generally comes larger false positive rates if the assumptions are not met

```
library(lme4)
```

```
## Loading required package: Matrix
```

```
#start with a random intercept for year
hier_pop_logit = glmer(intercon ~ country_pop +
                        aggdifxx + gdppc + polity2 +
                        (1|year),
                 data=data,
                 family = binomial)
```

```
## boundary (singular) fit: see ?isSingular
```

```
summary(hier_pop_logit) #so here we see that it is (near) singular, meaning we don't want to include th
```

```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: intercon ~ country_pop + aggdifxx + gdppc + polity2 + (1 | year)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##  10798.4  10840.5  -5393.2  10786.4     8210
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -6.3202 -1.0721  0.7150  0.8839  1.2244
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  year   (Intercept) 0        0
## Number of obs: 8216, groups:  year, 41
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.071172   0.058709  -1.212    0.225
## country_pop  0.451093   0.038043  11.857  < 2e-16 ***
## aggdifxx     0.037667   0.005560   6.775 1.25e-11 ***
## gdppc       -0.141813   0.035302  -4.017 5.89e-05 ***
## polity2      0.017915   0.003707   4.833 1.35e-06 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##             (Intr) cntry_ aggdfx gdppc
## country_pop  0.119
```

```
## aggdifxx    -0.918 -0.072
## gdppc        0.021 -0.005  0.044
## polity2      0.034 -0.166 -0.080 -0.452
## convergence code: 0
## boundary (singular) fit: see ?isSingular
```

```r
#perhaps the effect of country_pop varies by year
hier_pop_logit2 = glmer(intercon ~
                        aggdifxx + gdppc + polity2 +
                        (country_pop - 1|year),
                 data=data,
                 family = binomial)
summary(hier_pop_logit2)
```

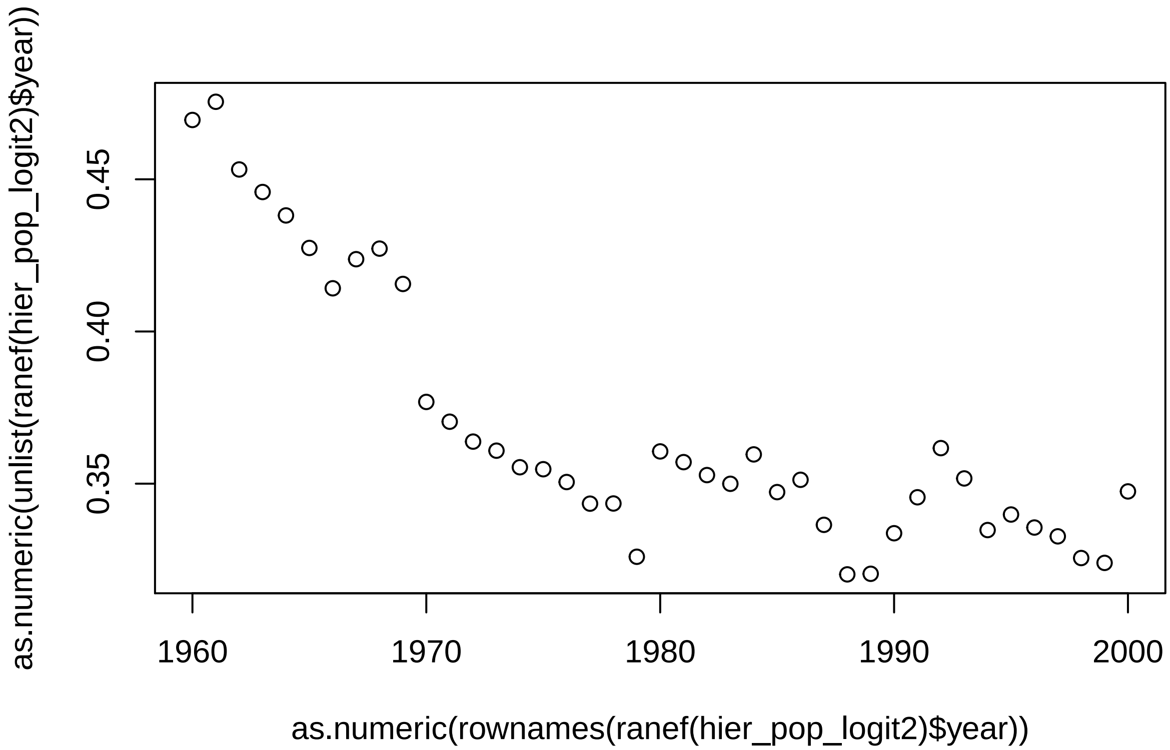
```
## Generalized linear mixed model fit by maximum likelihood (Laplace
##   Approximation) [glmerMod]
##  Family: binomial  ( logit )
## Formula: intercon ~ aggdifxx + gdppc + polity2 + (country_pop - 1 | year)
##    Data: data
##
##      AIC      BIC   logLik deviance df.resid
##  10898.6  10933.6  -5444.3  10888.6     8211
##
## Scaled residuals:
##     Min      1Q  Median      3Q     Max
## -4.5936 -1.0837  0.7195  0.8756  1.2190
##
## Random effects:
##  Groups Name        Variance Std.Dev.
##  year   country_pop 0.2039   0.4516
## Number of obs: 8216, groups:  year, 41
##
## Fixed effects:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.081469   0.058661  -1.389    0.165
## aggdifxx     0.038892   0.005557   6.998 2.59e-12 ***
## gdppc       -0.148490   0.035216  -4.217 2.48e-05 ***
## polity2      0.019613   0.003729   5.260 1.44e-07 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Correlation of Fixed Effects:
##          (Intr) aggdfx gdppc
## aggdifxx -0.918
## gdppc     0.025  0.042
## polity2   0.026 -0.073 -0.454
```

```r
ranef(hier_pop_logit2)
```

```
## $year
##      country_pop
## 1960   0.4694856
## 1961   0.4754654
## 1962   0.4532282
## 1963   0.4458256
```

```
## 1964   0.4381273
## 1965   0.4274407
## 1966   0.4141877
## 1967   0.4237514
## 1968   0.4272237
## 1969   0.4156131
## 1970   0.3768589
## 1971   0.3703605
## 1972   0.3638181
## 1973   0.3608440
## 1974   0.3553941
## 1975   0.3547446
## 1976   0.3505407
## 1977   0.3434365
## 1978   0.3434859
## 1979   0.3259891
## 1980   0.3605931
## 1981   0.3570795
## 1982   0.3528222
## 1983   0.3499583
## 1984   0.3596147
## 1985   0.3472264
## 1986   0.3512745
## 1987   0.3364797
## 1988   0.3202024
## 1989   0.3203892
## 1990   0.3337154
## 1991   0.3455315
## 1992   0.3616808
## 1993   0.3517183
## 1994   0.3347578
## 1995   0.3398619
## 1996   0.3355894
## 1997   0.3326971
## 1998   0.3255703
## 1999   0.3239733
## 2000   0.3474547
##
## with conditional variances for "year"
```

```r
plot(as.numeric(rownames(ranef(hier_pop_logit2)$year)), as.numeric(unlist(ranef(hier_pop_logit2)$year)))
```



Fractional Regression

- For proportions
- We can use a quasibinomial, but there are some undesirable properties of the estimator (such as the fact that proportions rarely follow the specified distribution)

```
library(frm)
#let's convert polity2 to a proportion and model with gdppc
data2 = na.omit(data[, c('polity2', 'gdppc')])
pol2 = (data2$polity2+10)/20
gdppc = as.matrix(data2$gdppc)
colnames(gdppc) = 'gdppc'
mod_frac = frm(pol2, gdppc, linkfrac = 'logit')
```

```
##
## *** Fractional logit regression model ***
##
##           Estimate Std. Error t value Pr(>|t|)
## INTERCEPT 0.289748   0.025498  11.364    0.000 ***
## gdppc     1.239066   0.078445  15.795    0.000 ***
##
## Note: robust standard errors
##
## Number of observations: 11240
## R-squared: 0.198
```

The Tobit Model

- When you have censoring in the outcome
- Can be on either or both sides

```r
#polity is actually censored (and discrete, but we'll ignore that)
library(AER)
```

```
## Loading required package: car

## Loading required package: carData

## Registered S3 methods overwritten by 'car':
##   method                          from
##   influence.merMod                lme4
##   cooks.distance.influence.merMod lme4
##   dfbeta.influence.merMod         lme4
##   dfbetas.influence.merMod        lme4

## Loading required package: lmtest

## Loading required package: zoo

##
## Attaching package: 'zoo'

## The following objects are masked from 'package:base':
##
##     as.Date, as.Date.numeric

## Loading required package: sandwich

## Loading required package: survival
```

```r
tob_mod = tobit(polity2 ~ gdppc, left = -10,
                right = 10, data = data)
summary(tob_mod)
```

```
##
## Call:
## tobit(formula = polity2 ~ gdppc, left = -10, right = 10, data = data)
##
## Observations: (5628 observations deleted due to missingness)
##          Total  Left-censored     Uncensored Right-censored
##          11240              0          11240              0
##
## Coefficients:
##             Estimate Std. Error z value Pr(>|z|)
## (Intercept) 1.792255   0.079208   22.63   <2e-16 ***
## gdppc       6.625863   0.140795   47.06   <2e-16 ***
## Log(scale)  2.033308   0.007585  268.06   <2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Scale: 7.639
##
## Gaussian distribution
## Number of Newton-Raphson Iterations: 4
## Log-likelihood: -3.357e+04 on 3 Df
```

```
## Wald-statistic:  2215 on 1 Df, p-value: < 2.22e-16
```

```r
#?tobit
```

Zero-Inflated Models

- We have already discussed the original zero-inflated logit, in which coefficient estimates (not including the intercept) will remain unchanged if you drop excess zeros
- More recent developments involve two-stage (but simultaneously estimated) regression
- Model the probability that the observation is an always zero vs. a potential non-zero
- Then multiply the probability of a potential non-zero with the distribution of interest
- Could be logit, Probit, Poisson, ZIMVOP, etc.
- Hurdle models also allow for undercount of zeros
- Again, we will cover in more detail in Bayesian weeks

```r
#Zero inflated Poisson for your rebellion data
library(pscl)
```

```
## Classes and Methods for R developed in the
## Political Science Computational Laboratory
## Department of Political Science
## Stanford University
## Simon Jackman
## hurdle and zeroinfl functions by Achim Zeileis
```

```r
#?zeroinfl
pop_zinf = zeroinfl(rebellion ~ country_pop + aggdifxx + gdppc + polity2,
                 data=data)
summary(pop_zinf)
```

```
##
## Call:
## zeroinfl(formula = rebellion ~ country_pop + aggdifxx + gdppc +
##     polity2, data = data)
##
## Pearson residuals:
##     Min      1Q  Median      3Q     Max
## -1.6775 -0.4889 -0.4610 -0.2641  5.1876
##
## Count model coefficients (poisson with log link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  0.797162   0.040073  19.893  < 2e-16 ***
## country_pop  0.028203   0.010041   2.809  0.00497 **
## aggdifxx     0.010105   0.003422   2.953  0.00315 **
## gdppc       -0.494643   0.049667  -9.959  < 2e-16 ***
## polity2     -0.021584   0.002475  -8.721  < 2e-16 ***
##
## Zero-inflation model coefficients (binomial with logit link):
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)  1.144376   0.076067  15.044  < 2e-16 ***
## country_pop -0.407026   0.028334 -14.365  < 2e-16 ***
## aggdifxx    -0.021375   0.006888  -3.103  0.00191 **
## gdppc       -0.255303   0.089160  -2.863  0.00419 **
## polity2      0.001996   0.004731   0.422  0.67315
```

```
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Number of iterations in BFGS optimization: 15
## Log-likelihood: -7946 on 10 Df
```

```r
#Zero inflated logit for your intercon
library(Zelig)
pop_z = zelig(intercon ~ country_pop +
                aggdifxx + gdppc + polity2,
              model = 'logit',
                 data=data)
```

```
## How to cite this model in Zelig:
##   R Core Team. 2007.
##   logit: Logistic Regression for Dichotomous Dependent Variables
##   in Christine Choirat, Christopher Gandrud, James Honaker, Kosuke Imai, Gary King, and Olivia Lau,
##   "Zelig: Everyone's Statistical Software," http://zeligproject.org/
```

```r
summary(pop_z) #this just drops zeros if needed, does not model two stages (need to move to writing you
```

```
## Model:
##
## Call:
## z5$zelig(formula = intercon ~ country_pop + aggdifxx + gdppc +
##     polity2, data = data)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.7248  -1.2371   0.9087   1.0745   1.3535
##
## Coefficients:
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept) -0.071172   0.058710  -1.212    0.225
## country_pop  0.451093   0.038043  11.857  < 2e-16
## aggdifxx     0.037667   0.005560   6.775 1.25e-11
## gdppc       -0.141813   0.035302  -4.017 5.89e-05
## polity2      0.017915   0.003707   4.833 1.35e-06
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 11183  on 8215  degrees of freedom
## Residual deviance: 10786  on 8211  degrees of freedom
##   (8652 observations deleted due to missingness)
## AIC: 10796
##
## Number of Fisher Scoring iterations: 5
##
## Next step: Use 'setx' method
```