# Multilevel Models

## 11. Models for Ordinal Data

Germán Rodríguez

Princeton University

April 30, 2018

# Categorical Data

Our final unit concerns models for categorical data. We will consider ordered logit models first, which are simpler, and then turn our attention to multinomial logit models.

MALMUS notes that at the time of writing there were no official Stata commands for fitting multilevel models for categorical data, but version 14 solved the problem for ordered logits with `meologit`. As for multinomial logit models, it turns out that they can be fit as structural equation models with `gsem`, as noted by a Stata blogger.

On the R ecology I haven't found any package to fit multilevel ordered or multinomial logit models by maximum likelihood, but there are plenty of Bayesian solutions. We will use this opportunity to gather a bit more experience using Stan.

# Ordered Logit Models

Recall than in an ordered logit model we focus on the logit of *cumulative* probabilities, so given an outcome $Y_{ij}$ for the $j$-th observation in group $i$ a random-intercept model would be

$$\Pr\{Y_{ij}|a_i > k\} = \text{logit}^{-1}(a_i + \boldsymbol{x}'_{ij}\boldsymbol{\beta} - \theta_k)$$

where $a_i \sim N(0, \sigma_a^2)$ is a normally-distributed random effect with mean 0 and variance $\sigma_a^2$.

The model may also be written in terms of a latent variable following a linear model

$$Y^*_{ij} = a_i + \boldsymbol{x}'_{ij}\boldsymbol{\beta} + e_{ij}$$

where $e_{ij}$ is standard logistic and $Y_{ij} > k \iff Y^*_{ij} > \theta_k$, so the $\theta$'s may be interpreted as threshold parameters.

The equivalence follows from substituting the latent variable in $\Pr\{Y^*_{ij} > \theta_k\}$ and using the symmetry of the logistic distribution.

# Treating Schizophrenia

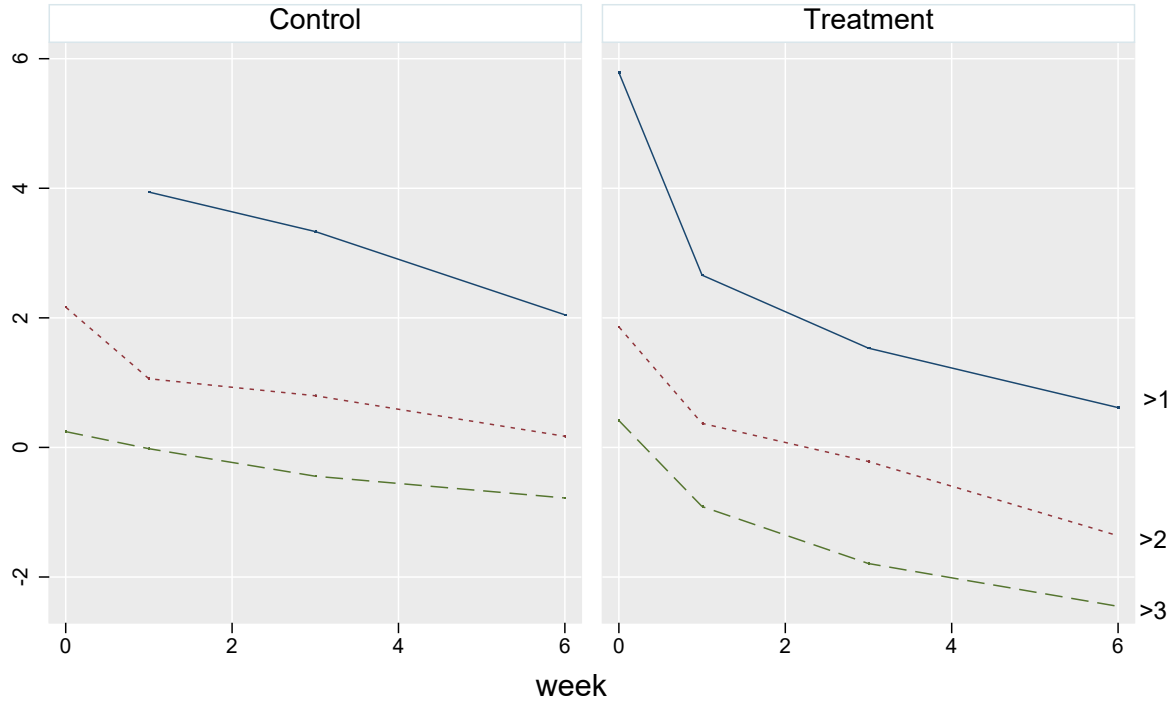
We'll analyze the example in MALMUS, a randomized trial comparing four drugs and a placebo and measuring the severity of illness using the Inpatient Multidimensional Psychiatric Scale (IMPS) at various intervals since randomization.

We combine all four drugs in a single "treated" group and recode the outcome into four severity categories: normal or borderline ($\leq 2.4$), moderately ill ($2.5 - 4.4$), markedly ill ($4.5 - 5.4$) and severely ill ($5.5 - 7$), as done in the original analysis.

As always, it pays to examine the data before analysis. Patients can be seen for up to seven weeks, but the most common pattern has observations in weeks 0, 1, 3 and 6. In fact no patient has more than 4 assessments.

# Plotting Cumulative Proportions

A useful diagnostic plot shows the empirical logits of the proportions above each response category by week. Because weeks 2, 4 and 5 have very few assessments we omit them from the plot.




The graph shows that the treatment is generally beneficial but the trajectories are not linear. We will follow the original authors and work with the square root of weeks as the time scale.

# Ordered Logits

Obviously we will need to interact treatment and time to capture treatment effects on the trajectory of each patient.

Here is a baseline ordered logit model representing population average effects (with uncorrected standard errors)

```
Ordered logistic regression                     Number of obs     =      1,603
                                                LR chi2(3)        =     501.26
                                                Prob > chi2       =     0.0000
Log likelihood = -1878.0969                     Pseudo R2         =     0.1177

------------------------------------------------------------------------------
       impso |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    sqrtweek |  -.5366467    .110815    -4.84   0.000    -.7538401   -.3194534
   treatment |  -.0006043   .1883287    -0.00   0.997    -.3697218    .3685132
 interaction |  -.7509692   .1276787    -5.88   0.000    -1.001215   -.5007235
-------------+----------------------------------------------------------------
       /cut1 |  -3.807279   .1898591                     -4.179396   -3.435162
       /cut2 |  -1.760167   .1702695                     -2.093889   -1.426445
       /cut3 |  -.4221112   .1636329                     -.7428258   -.1013965
------------------------------------------------------------------------------
```

# Random-Intercept Ordered Logits

Next we add a patient-specific random intercept, assumed independent of the covariates across patients.

```
meologit impso weeksqrt treatment interact || id:

Mixed-effects ologit regression                 Number of obs     =      1,603
Group variable:                 id              Number of groups  =        437
 ...
Integration method: mvaghermite                 Integration pts.  =          7

                                                Wald chi2(3)      =     480.06
Log likelihood = -1701.3811                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
       impso |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
    weeksqrt |  -.7657629   .1307697    -5.86   0.000    -1.022067    -.509459
   treatment |  -.0603847   .3136873    -0.19   0.847    -.6752006    .5544311
    interact |  -1.206126   .1526656    -7.90   0.000    -1.505345   -.9069068
-------------+----------------------------------------------------------------
       /cut1 |  -5.860997   .3321236   -17.65   0.000    -6.511947   -5.210046
       /cut2 |  -2.828207   .2901595    -9.75   0.000     -3.39691   -2.259505
       /cut3 |  -.7103887   .2749679    -2.58   0.010    -1.249316   -.1714614
-------------+----------------------------------------------------------------
id           |
   var(_cons)|   3.773713   .4650158                      2.964009     4.80461
------------------------------------------------------------------------------
LR test vs. ologit model: chibar2(01) = 353.43        Prob >= chibar2 = 0.0000
```

This model yields an intra-class correlation of 0.53 in the latent scale.

# Interpreting Random Intercept Results

The treatment coefficient reflects initial differences and it is reassuringly small and not significant.

The interesting coefficient is the interaction, which exponentiated is 0.299. This indicates that the odds of begin above category 1, 2 or 3 of the IMPS are 70% lower in the treatment than in the control group at any week after randomization.

The standard deviation of the random effect indicates very substantial variation across patients, with the odds of being above any category increasing seven-fold as we move up one standard deviation from the mean with everything else the same.

We can also compute a median odds ratio $\exp\{\sqrt{2}\sigma_a\Phi^{-1}(3/4)\}$ as 6.37. This means that if we draw at random two patients with the same covariates, the ratio of the odds of scoring above any given category, when we compare the larger to the smaller odds, would exceed 6.37 half the time.

# Random-Slope Ordered Logits

The next model allows the slope of the time variable to vary randomly across patients. As usual we specify an unstructured covariance matrix.

```
meologit impso weeksqrt treatment interact || id: weeksqrt, covariance(unstructured)
...
                                                Wald chi2(3)      =     254.29
Log likelihood =   -1662.73                     Prob > chi2       =     0.0000
------------------------------------------------------------------------------
             impso |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------------+----------------------------------------------------------------
          weeksqrt |  -.8821765   .2175176    -4.06   0.000    -1.308503   -.4558499
         treatment |   .0525632   .3898986     0.13   0.893    -.7116241    .8167505
          interact |  -1.695097   .2520524    -6.73   0.000    -2.189111   -1.201084
-------------------+----------------------------------------------------------------
             /cut1 |   -7.32517   .4727348   -15.50   0.000    -8.251714   -6.398627
             /cut2 |  -3.423091   .3857357    -8.87   0.000    -4.179119   -2.667062
             /cut3 |  -.8174723   .3506013    -2.33   0.020    -1.504638   -.1303064
-------------------+----------------------------------------------------------------
id                 |
      var(weeksqrt)|   2.009688   .4179082                      1.336977    3.020879
         var(_cons)|   6.993466   1.313759                      4.839381    10.10637
-------------------+----------------------------------------------------------------
id                 |
cov(_cons,weeksqrt)|  -1.504658   .5300824    -2.84   0.005      -2.5436   -.4657153
------------------------------------------------------------------------------
LR test vs. ologit model: chi2(3) = 430.73                Prob > chi2 = 0.0000
```

# Interpreting Random Slope Results

A comparison with the previous model yields a chi-squared of 77.24. Although the test is conservative (because we are on a boundary of the parameter space) it is clearly highly significant.

The patient-specific odds ratio per unit of time is estimated as 0.41 in the control group and 0.07 in the treated group. Both the intercept and slope vary substantially across patients with a correlation of $-0.40$.

As MALMUS notes, this means that patients having more severe schizophrenia at the start of the study tend to have a greater decline in severity than those with less severe schizophrenia in both the control and treatment groups.

We'll leave as an exercise computing subject-specific and population-average predicted probabilities by treatment and week.

# Fitting the Models in R

We now fit exactly the same models in R. I will not repeat the graphs, but note that we can fit the standard proportional odds logistic regression model using the function `polr` in the `MASS` package. Given a data frame called `sch` the call is:

```
podds <- polr(impso ~ weeksqrt * treatment, data = sch)
 ...
> summary(podds)
...
Coefficients:
                        Value Std. Error   t value
weeksqrt           -0.5366419     0.1108 -4.842684
treatment          -0.0005995     0.1883 -0.003183
weeksqrt:treatment -0.7509752     0.1277 -5.881755

Intercepts:
                    Value    Std. Error t value
(0,2.4]|(2.4,4.4]    -3.8073   0.1899   -20.0532
(2.4,4.4]|(4.4,5.4]  -1.7602   0.1703   -10.3375
(4.4,5.4]|(5.4,7]    -0.4221   0.1636    -2.5796

Residual Deviance: 3756.194
AIC: 3768.194
```

It is reassuring to see that we have the same results as in Stata. We now try Stan.

# Ordered Logit Model in Stan

We'll build the model in steps, starting from the standard ordered logit model.

```
sch_code = '
data {
  int N;  // number of observations
  int K;  // number of response categories
  int D;  // number of predictors
  int<lower=1, upper=K> y[N]; // outcomes
  row_vector[D] x[N];    // predictors
}
parameters {
  ordered[K-1] theta;
  vector[D] beta;
}
model {
  for(n in 1:N) {
    y[n] ~ ordered_logistic(x[n] * beta, theta);
  }
}'
```

The code follows the Stan manual and is remarkably simple thanks to the fact that there is an `ordered` data type to handle the thresholds and an `ordered_logistic` distribution to take care of converting the tail probabilities into a multinomial distribution.

# Bayesian Ordered Logit Estimates

The next step was to put the data in a list and run Stan

```
sch_data <- list(N = nrow(sch), K = 4, D = 3,
  y = as.numeric(sch$impso), x = as.matrix(sch[,c("treatment","weeksqrt","interaction")]))
ologit <- stan(model_code=sch_code, model_name="ologit", data=sch_data, iter=2000, chains=2)
```

I specified a few options to print the results in a convenient way

```
> print(ologit, digits_summary=3, probs=c(0.025,0.5,0.975))
Inference for Stan model: ologit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

              mean se_mean    sd      2.5%       50%     97.5% n_eff  Rhat
theta[1]    -3.820   0.008 0.191    -4.196    -3.825    -3.445   572 1.000
theta[2]    -1.766   0.007 0.173    -2.084    -1.769    -1.419   554 0.999
theta[3]    -0.423   0.007 0.167    -0.742    -0.424    -0.080   528 1.000
beta[1]      0.004   0.008 0.193    -0.364    -0.003     0.389   518 1.000
beta[2]     -0.537   0.005 0.111    -0.739    -0.538    -0.315   554 0.999
beta[3]     -0.757   0.005 0.129    -1.011    -0.757    -0.507   555 1.000
lp__     -1880.036   0.065 1.687 -1884.097 -1879.734 -1877.659   671 1.004

Samples were drawn using NUTS(diag_e) at Sat Apr 23 14:47:54 2016.
```

The Bayesian estimates are very similar to the maximum likelihood estimates obtained earlier, so we soldier on.

# Specifying a Random Intercept Model

Things get more interesting when we add a random intercept at the patient level. We assume that $a_i \sim N(0, \sigma)$ with a $U(0, 100)$ prior on $\sigma$ and the default priors on everything else.

```
    sch_code = '
    data {
      int N;  // number of observations
|     int M;  // number of groups
      int K;  // number of response categories
      int D;  // number of predictors

      int<lower=1, upper=K> y[N]; // outcomes
      row_vector[D] x[N];         // predictors
|     int g[N];                   // map observations to groups
    }
    parameters {
      ordered[K-1] theta;
      vector[D] beta;
|     real a[M];
|     real<lower=0, upper=10> sigma;
    }
    model {
|     a ~ normal(0, sigma);
      for(n in 1:N) {
|       y[n] ~ ordered_logistic(x[n] * beta + a[g[n]], theta);
      }
    }'
```

A bar on the left margin marks new or changed lines.

# Additions for Random Intercept Model

The changes to the code include

- adding the number of groups and a map to the data block
- adding the group random effects and $\sigma_a$ to the parameters
- defining the prior for the random effects and modifying the linear predictor

The code assumes that the group id's are consecutive integers, which is not the case in this dataset. I wrote the following general function to map group id's when they are not the integers `1:M`:

```
map_groups <- function(id) {
   f <- table(id)
   rep(1:nrow(f), f)
 }
```

And we can then add the map to the list

```
sch_data$g = map_groups(sch$id))
```

# Running the Random Intercept Model

We can now run the model and (eventually) print the results. I specify the parameters to be printed to omit the random effects

```
riologit <- stan(model_code=sch_code, model_name="riologit", data=sch_data, iter=2000, chains=2)
...
print(riologit, digits_summary=3, probs=c(0.025,0.5,0.975),
      pars=c("theta[1]","theta[2]","theta[3]","beta[1]","beta[2]","beta[3]","sigma"))
Inference for Stan model: riologit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

           mean se_mean    sd   2.5%    50%  97.5% n_eff  Rhat
theta[1] -5.882   0.018 0.329 -6.553 -5.873 -5.273   351 1.007
theta[2] -2.834   0.015 0.290 -3.420 -2.822 -2.288   383 1.004
theta[3] -0.703   0.013 0.273 -1.251 -0.694 -0.199   433 1.003
beta[1]  -0.771   0.005 0.130 -1.032 -0.772 -0.519   629 1.000
beta[2]  -0.043   0.015 0.308 -0.651 -0.035  0.530   409 1.003
beta[3]  -1.210   0.006 0.150 -1.503 -1.208 -0.915   549 1.002
sigma     1.965   0.007 0.119  1.741  1.963  2.205   287 1.014

Samples were drawn using NUTS(diag_e) at Sat Apr 23 15:13:58 2016.
For each parameter, n_eff is a crude measure of effective sample size,
and Rhat is the potential scale reduction factor on split chains (at
convergence, Rhat=1).
```

One again the results are very similar to the maximum likelihood estimates, so we are encouraged to continue.

# Specifying a Random Slope Ordered Logit Model

The final step is to add a random slope. Here's the new code:

```
sch_code = '
data {
  int N;  // number of observations
  int M;  // number of groups
  int K;  // number of response categories
  int D;  // number of predictors

  int<lower=1, upper=K> y[N]; // outcomes
  row_vector[D] x[N];         // predictors
  int g[N];                   // map observations to groups
  vector[2] Zero;             // means of random effects
}
parameters {
  ordered[K-1] theta;
  vector[D] beta;
  vector[2] u[M];
  corr_matrix[2] Omega;
  vector<lower=0>[2] sigma;
}
transformed parameters {
  cov_matrix[2] Sigma;
  Sigma <- quad_form_diag(Omega, sigma);
}
model {
  u ~ multi_normal(Zero, Sigma);
  for(n in 1:N) {
    y[n] ~ ordered_logistic(x[n] * beta +
      u[g[n]][1] + u[g[n]][2]*x[n][1], theta);
  }
}'
```

# Additions for Random Slope Ordered Logit Model

The basic idea is that we now have bivariate normal random effects

$$u = \begin{pmatrix} a \\ b \end{pmatrix} \sim N_2 \left( \begin{pmatrix} 0 \\ 0 \end{pmatrix}, \begin{pmatrix} \sigma_a^2, \sigma_{ab} \\ \sigma_{ab}, \sigma_b^2 \end{pmatrix} \right)$$

with an unstructured covariance matrix. One way to parametrize the variance-covariance matrix is in terms of non-negative standard deviations $\sigma_a, \sigma_b$ and a correlation matrix, which is what we do with `sigma` and `Omega`.

We then define a *transformed parameter* to obtain the 2x2 covariance matrix `Sigma`, which can be computed from the standard deviations and correlations using the function `quad_form_diag()`.

All that remains then is to sample the bivariate random effects from a multivariate normal distribution and add them to the linear predictor, remembering to multiply the slope by the time variable.

# Running the Random Slope Ordered Logit Model

We add a vector of zeroes to the data and run the model

```
sch_data$Zero <- c(0,0)
rsologit <- stan(model_code=sch_code, model_name="rsologit", data=sch_data, iter=2000, chains=2)
```

When it's all done we print the results

```
Inference for Stan model: rsologit.
2 chains, each with iter=2000; warmup=1000; thin=1;
post-warmup draws per chain=1000, total post-warmup draws=2000.

             mean se_mean    sd   2.5%    50%  97.5% n_eff  Rhat
theta[1]   -7.454   0.034 0.489 -8.415 -7.443 -6.542   211 1.008
theta[2]   -3.485   0.018 0.393 -4.255 -3.477 -2.726   457 1.004
theta[3]   -0.839   0.013 0.359 -1.553 -0.825 -0.120   792 1.001
beta[1]    -0.892   0.008 0.221 -1.308 -0.892 -0.465   808 1.002
beta[2]     0.055   0.013 0.399 -0.713  0.060  0.882   965 1.000
beta[3]    -1.735   0.008 0.253 -2.227 -1.732 -1.234   941 1.000
Sigma[1,1]  7.482   0.129 1.381  5.067  7.411 10.453   114 1.016
Sigma[1,2] -1.647   0.055 0.558 -2.872 -1.616 -0.671   102 1.018
Sigma[2,1] -1.647   0.055 0.558 -2.872 -1.616 -0.671   102 1.018
Sigma[2,2]  2.198   0.049 0.470  1.364  2.169  3.213    93 1.012

Samples were drawn using NUTS(diag_e) at Sat Apr 23 17:03:40 2016.
```

One more time the results are similar to the maximum likelihood estimates.

# Trace Plots for Random Slope Model

```
traceplot(rsologit, pars=c("theta[1]","theta[2]","theta[3]","beta[1]","beta[2]","beta[3]",
    "Sigma[1,1]","Sigma[2,2]","Sigma[1,2]"))
```