Multilevel Models

12. Models for Nominal Data

Germán Rodríguez

Princeton University

May 2, 2018

Multinomial Logit Model

Recall the multinomial logit model, where the probability of falling in category k for individual i is

$$\Pr\{Y_i = k\} = \frac{e^{\boldsymbol{x}_i'\boldsymbol{\beta}_k}}{\sum_v e^{\boldsymbol{x}_i'\boldsymbol{\beta}_v}}$$

To identify the model we choose a category as reference and set $\beta_r = 0$ so the model has $K - 1$ sets of coefficients and is identified.

The k-th linear predictor $\boldsymbol{x}_i'\boldsymbol{\beta}_k$ is the log-relative probability of category k relative to r (also called the log-odds of k over r).

The model can be interpreted in terms of random utilities where the utility of choice k for individual i follows the linear model

$$U_{ik} = \boldsymbol{x}_i'\boldsymbol{\beta}_k + e_{ik}$$

where the e_{ik} are i.i.d. extreme value and $U_{ir} = 0$ serves as a baseline. Maximizing the expected utility leads to choosing category k with the probability given above.

Random-Intercept Multinomial Logits

We now extend the model to two-level data so Y_{ij} is the outcome for individual j in group i. We introduce K random intercepts per individual, so the conditional probability of falling in category k is

$$\Pr\{Y_i = k | \boldsymbol{a}_i\} = \frac{e^{a_{ik} + \boldsymbol{x}_i' \boldsymbol{\beta}_k}}{\sum_v e^{a_{iv} + \boldsymbol{x}_i' \boldsymbol{\beta}_v}}$$

but set $a_r = 0$ so we are left with $K - 1$ random effects assumed to have a multivariate normal distribution with mean vector zero and arbitrary variance-covariance matrix.

The log-relative conditional probability of category k over r given the random effects a_{ik} for $k \neq r$ is then

$$\log \frac{\Pr\{Y_{ik} | \boldsymbol{a}_i\}}{\Pr\{Y_{ir} | \boldsymbol{a}_i\}} = a_{ik} + \boldsymbol{x}_i' \boldsymbol{\beta}_k$$

so a_{ik} can be interpreted as a latent propensity to choose category k over r net of the covariates.

The McKinney Homeless Study

We will illustrate the methods using the McKinney Homeless study, which has generated interesting longitudinal data on 361 at-risk individuals randomly assigned to one of two types of case management (comprehensive vs. traditional) and one of two levels of access to independent housing using "Section 8" certificates.

The outcome is housing status at baseline and at 6, 12 and 24 months, classified as streets/shelters, community housing, or independent housing. The predictors of interest include time and `sec8`, a dummy variable coded one for the treatment group.

The data have been analyzed by Don Hedeker, author of the `mixno` package* for fitting mixed multinomial models using Gauss-Hermite quadrature. We will fit essentially the same model, although for simplicity we will treat time (coded 0 to 3) linearly instead of using dummy variables.

*`https://www.jstatsoft.org/article/view/v004i05`

Population Average Effects

For reference purposes I fitted a standard multinomial logit model estimating population average effects (with uncorrected standard errors).

```
. mlogit status c.sec8##c.time,  base(0)
Multinomial logistic regression                 Number of obs     =      1,289
                                                LR chi2(6)        =     316.57
                                                Prob > chi2       =     0.0000
Log likelihood =   -1223.16                     Pseudo R2         =     0.1146
```

status	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
street	(base outcome)					
community						
sec8	.2395584	.2130131	1.12	0.261	-.1779395	.6570564
time	.7961881	.1075957	7.40	0.000	.5853045	1.007072
c.sec8#c.time	-.5180044	.1576099	-3.29	0.001	-.8269142	-.2090947
_cons	-.2109418	.1428502	-1.48	0.140	-.4909231	.0690395
independent						
sec8	1.157348	.2551718	4.54	0.000	.6572208	1.657476
time	1.138287	.123074	9.25	0.000	.8970665	1.379508
c.sec8#c.time	-.2308719	.1637933	-1.41	0.159	-.5519008	.090157
_cons	-1.405489	.1964876	-7.15	0.000	-1.790598	-1.02038

We'll compare these to Bayes estimates.

Maximum Likelihood via SEM

Stata does not have a command for multinomial logit models with random effects, but Rebecca Pope explains how to fit the model using structural equation models via gsem in the Stata Newsletter http://www.stata.com/stata-news/news29-2/xtmlogit/ using

```
gsem (1.status <- sec8 time secXtime RI1[id]) ///
     (2.status <- sec8 time secXtime RI2[id]), mlogit}
```

The model defines two latent variables that vary across groups to capture random effects for each equation. The variances and covariance of these random effects are

var(RI1[id])	1.744592	.4753894			1.022697	2.976052
var(RI2[id])	3.818815	.7779019			2.56175	5.692728
cov(RI2[id],RI1[id])	1.701683	.5029326	3.38	0.001	.7159536	2.687413

This implies a correlation of 0.66 between the two latent variables representing the contrast of community over street and of independent over street.

Subject-Specific Effects

The estimates of the fixed effects are shown below

	Coef.	Std. Err.	z	P>|z|	[95% Conf.	Interval]
0.status	(base outcome)					
1.status						
sec8	.3835373	.2829853	1.36	0.175	-.1711037	.9381783
time	1.015074	.1329061	7.64	0.000	.7545826	1.275565
secXtime	-.5786798	.181297	-3.19	0.001	-.9340153	-.2233442
RI1[id]	1	(constrained)				
_cons	-.2063278	.1926945	-1.07	0.284	-.5840022	.1713466
2.status						
sec8	1.60725	.3791572	4.24	0.000	.8641157	2.350384
time	1.530787	.1579142	9.69	0.000	1.221281	1.840293
secXtime	-.2223493	.1998801	-1.11	0.266	-.6141072	.1694085
RI2[id]	1	(constrained)				
_cons	-2.047767	.3034708	-6.75	0.000	-2.642559	-1.452975

There is a trend away from the street, towards community housing in the control group and towards independent housing in the Section 8 group.

Multinomial Logit Models via Stan

Let us now explore fitting these models in a Bayesian framework using Stan. We start with the standard multinomial logit model.

There is an example in the Stan manual using one equation per outcome, a model that they note is identified only "if there are suitable priors on the coefficients". A faster and in my view preferable alternative is to work with only $K - 1$ equations for K response categories, as we did for maximum likelihood.

We define the coefficients to be estimated as a $K - 1$ by P matrix, and then add a row of zeroes to match the reference category in a new K by P matrix defined in the transformed parameters block.

The function used to convert multinomial logits to probabilities is called `softmax` in the machine learning literature and Stan. The newer function `categorical_logit()` calls that implicitly.

Stan Code for Multinomial Logit

```
sd_model <- '
  data {
    int K;   // number of outcome categories
    int K1;  // K-1
    int N;   // number of observations
    int P;   // number of predictors a.k.a. D
    int y[N];// outcome, coded 1 to K for each obs
    vector[P] x[N]; // predictors, including constant
  }
  transformed data {
    row_vector[P] base;
    base = rep_row_vector(0, P);
  }
  parameters {
    matrix[K1,P] beta;
  }
  transformed parameters {
    matrix[K, P] betap;
    betap = append_row(base, beta);
  }
  model {
    // prior for beta (vectorized)
    for(k in 1:K1) {
      beta[k] ~ normal(0,5);
    }
    // likelihood of outcome
    for(n in 1:N) {
      y[n] ~ categorical_logit(betap * x[n]);
    }
  }
'
```

The Stan Output

And here are the results of running the model

```
mlogit <- stan(model_code=sd_model,model="mlogit",data=sd_data,iter=2000,chains=2)

> print(mlogit, pars="beta", digits_summary=3, probs=c(0.025,0.5,0.975))
Inference for Stan model: mlogit.
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
beta[1,1] -0.216   0.005 0.148 -0.503 -0.216  0.070   872 1.001
beta[1,2]  0.246   0.007 0.218 -0.184  0.249  0.662   966 1.000
beta[1,3]  0.805   0.004 0.109  0.602  0.804  1.026   933 1.004
beta[1,4] -0.525   0.005 0.159 -0.832 -0.526 -0.213   919 1.003
beta[2,1] -1.421   0.007 0.201 -1.835 -1.423 -1.044   724 1.000
beta[2,2]  1.171   0.009 0.263  0.669  1.174  1.684   878 1.000
beta[2,3]  1.151   0.005 0.124  0.906  1.154  1.386   669 1.002
beta[2,4] -0.239   0.006 0.166 -0.558 -0.241  0.086   830 1.002

Samples were drawn using NUTS(diag_e) at Tue May 01 13:47:57 2018.
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).
```

The measures of effective sample size are all reassuring and the values of Rhat are close to 1 as they should be at convergence.

Comparison of Maximum Likelihood and Bayes

Here's a side-by-side comparison of ML and Bayes estimates by equation

Variable	Community/Street		Independent/Street	
Name	ML	Bayes	ML	Bayes
sec8	0.240	0.246	1.157	1.171
time	0.796	0.805	1.138	1.151
interaction	-0.518	-0.525	-0.231	-0.239
constant	-0.211	-0.216	-1.405	-1.421

I think the agreement is quite remarkable. We are thus encouraged to try adding random effects.

Random Intercepts in Multinomial Logits

We will add two correlated random intercepts for each individual, representing unobserved effects on the propensity to be in community and in independent housing rather than on the street. To define the model I generally followed the Stan manual.

We will define the multivariate normal distribution in terms of a vector of scale parameters and a matrix of correlations, which are the actual parameters to be estimated, just as we did for the random slope ordered logit model. This time, however, I defined the vector of means in a `transformed data` block.

In the model block we define the priors and hyper-priors for the random effects. The random effects are `multi_normal`. For the scales of the random effects I tried half `cauchy(0, 2.5)` priors, but got better results with uniforms. For the correlation I used a LKJ prior with parameter 2; for more on this prior see `http://www.psychstatistics.com/2014/12/27/d-lkj-priors/`.

Stan Code for Random-Effects Multinomial Logit

The model is long enough that I will present it in two parts. Here's Part 1 showing the data, transformed data and parameters.

```
sd_model <- '
    data {
      int K;          // number of outcome categories
      int K1;         // K-1
      int N;          // number of observations
      int P;          // number of predictors a.k.a. D
      int y[N];       // outcome, coded 1 to K for each obs
      vector[P] x[N]; // predictors, including constant
      int G;          // number of groups
      int map[N];     // map obs to groups
    }
    transformed data {
      vector[K1] zero;
      real baseline;
      zero = rep_vector(0, K1);
      baseline = 0;
    }
    parameters {
      matrix[K1,P] beta;         // fixed effects
      corr_matrix[K1] omega;     // ranef correlations
      vector<lower=0,upper=10>[K1] sigma; // ranef scales
      vector[K1] u[G];           // random intercepts
    }
...
```

Stan Code for Random-Effects Multinomial Logit

And here's Part 2, showing transformed parameters and the model block:

```
transformed parameters{
  cov_matrix[K1] V;
  V = quad_form_diag(omega, sigma);
}
model {
  // prior for beta (vectorized)
  for(k in 1:K1) {
    beta[k] ~ normal(0,5);
  }
  // prior/hyper prior for random effects
  // sigma ~ cauchy(0, 2.5);
  omega ~ lkj_corr(2);
  for(g in 1:G) {
    u[g] ~ multi_normal(zero, V);
  }
  { // local block for linear predictor
    vector[K] xb;
    for(n in 1:N) {
      xb = append_row(baseline, beta*x[n] + u[map[n]]);
      y[n] ~ categorical_logit(xb);
    }
  }
}'
```

The local block is used to add a zero to the linear predictor.

This is the call used to run the model

```
rimlogit <- stan(model_code=sd_model,model="rimlogit",data=sd_data,iter=2000,chains=2)
```

And these are the results

```
> print(rimlogit, digits_summary=3, probs=c(0.025,0.5,0.975),
+       pars=c("beta","sigma","omega[1,2]"))
Inference for Stan model: rimlogit.
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
beta[1,1]  -0.208   0.004 0.197 -0.602 -0.207  0.175  2000 0.999
beta[1,2]   0.363   0.006 0.291 -0.193  0.357  0.944  2000 1.001
beta[1,3]   1.018   0.005 0.133  0.770  1.016  1.281   865 1.001
beta[1,4]  -0.583   0.004 0.175 -0.919 -0.587 -0.252  2000 1.001
beta[2,1]  -2.056   0.008 0.298 -2.633 -2.048 -1.477  1260 1.000
beta[2,2]   1.584   0.009 0.371  0.872  1.576  2.301  1566 1.000
beta[2,3]   1.535   0.005 0.156  1.236  1.530  1.852  1064 1.000
beta[2,4]  -0.215   0.004 0.194 -0.616 -0.215  0.142  2000 1.000
sigma[1]    1.339   0.015 0.196  0.943  1.333  1.732   172 1.008
sigma[2]    1.965   0.011 0.198  1.620  1.957  2.378   322 1.002
omega[1,2]  0.625   0.006 0.092  0.427  0.632  0.778   254 1.000

Samples were drawn using NUTS(diag_e) at Tue May 01 10:58:04 2018.
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).
```

Comparison of Estimates

Here are the maximum likelihood and Bayesian estimates

Variable	Community/Street		Independent/Street	
Name	ML	Bayes	ML	Bayes
sec8	0.384	0.363	1.620	1.584
time	1.015	1.018	1.536	1.535
interaction	-0.579	-0.583	-0.220	-0.215
constant	-0.206	-0.208	-2.079	-2.056
scale	1.321	1.339	1.954	1.975
correlation	0.659	0.625		

The two sets of estimates are remarkably close, as one would expect from generally non-informative priors.

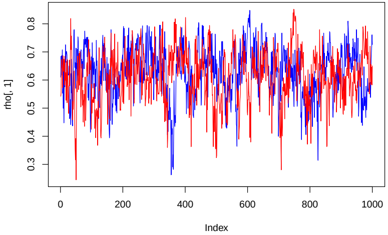
I report the posterior means of the scale parameters and correlation coefficient rather than the variances and covariance, so for comparability I translated the maximum likelihood results.

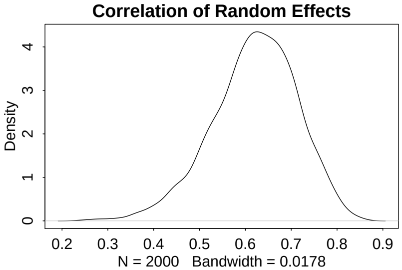
Trace Plots and Posterior Densities

We can extract any coefficient using `extract`. Here's the correlation of the random effects

```
r <- as.data.frame(extract(rimlogit, pars="omega[1,2]"), permute=FALSE)
```

And we can then do trace and/or density plots





We have a nice fuzzy caterpillar and the posterior is fairly symmetric around the mean of 0.6.

Calculating Predicted Probabilities

We can also use the samples to calculate any function of interest. Here's code to compute the predicted probabilities for the average person at time 3 under control and "Section 8" conditions.

```
> ep <- as.data.frame(extract(rimlogit, "beta"))
> names(ep)<-c("cons1","cons2","sec1","sec2","time1","time2","int1","int2")
> u0 =  cbind(0, ep[,"cons1"]+ep[,"time1"]*3, ep[,"cons2"]+ep[,"time2"]*3)
> p0 = colMeans(exp(u0)/rowSums(exp(u0)))
> u1 = u0 + cbind(0, ep[,"sec1"]+ep[,"int1"]*3, ep[,"sec2"]+ep[,"int2"]*3)
> p1 = colMeans(exp(u1)/rowSums(exp(u1)))
> rbind(p0,p1)
         [,1]      [,2]      [,3]
p0 0.03376637 0.5530774 0.4131562
p1 0.02773570 0.1153388 0.8569255
```

The probability of being in independent housing at the end of follow up for the average person is 41% in the control group and 86% in the Section 8 group, with only 3% on the street.

Try doing a trace and/or density plot, or constructing a 95% credible interval for the probability of independent housing.