POLI 7050
Spring 2008

March 5, 2008
Unordered Response Models II

Introduction

Today we'll talk about interpreting MNL and CL models. We'll start with general issues of model fit, and then get to variable effects. Note that nearly everything we do for MNLs also applies to CLs; accordingly, we'll focus more on the former, though we'll walk through a conditional logit example at the end as well.

We'll once again use the 1992 election as a running example. The data are 1473 voting respondents from the 1992 National Election Study, and the response (dependent) variable is who each respondent voted for, coded one for Bush (the elder), two for Clinton, and three for Perot. The basic model we'll be interpreting is an extension of the one we used in class last week; it looks like this:

```
. mlogit presvote partyid age white female, basecategory(1)


Multinomial logistic regression                   Number of obs   =       1473
                                                  LR chi2(8)      =     951.58
                                                  Prob > chi2     =     0.0000
Log likelihood = -1053.6506                       Pseudo R2       =     0.3111

------------------------------------------------------------------------------
    presvote |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
2            |
     partyid |  -1.135615   .0548618   -20.70   0.000    -1.243142   -1.028088
         age |  -.0026013   .0051396    -0.51   0.613    -.0126746     .007472
       white |    -.98908   .3134669    -3.16   0.002    -1.603464   -.3746961
      female |   -.125005   .1689499    -0.74   0.459    -.4561406    .2061307
       _cons |   5.806651   .4430144    13.11   0.000     4.938358    6.674943
-------------+----------------------------------------------------------------
3            |
     partyid |  -.5013218   .0486977   -10.29   0.000    -.5967675   -.4058761
         age |   -.015565   .0050436    -3.09   0.002    -.0254503   -.0056796
       white |   .8791807   .4360556     2.02   0.044     .0245275    1.733834
      female |   -.509278   .1626614    -3.13   0.002    -.8280884   -.1904676
       _cons |   1.980081   .5245439     3.77   0.000     .9519936    3.008168
------------------------------------------------------------------------------
(presvote==1 is the base outcome)
```

Model Fit in the MNL Model

First, we'll discuss overall model "fit." Consider again a general model for $i = \{1, ...N\}$ observations where we have J possible outcomes $j \in \{1, ...J\}$ on the dependent variable Y_i, and K independent variables $X_{ik} \in \{X_{i1}...X_{iK}\}$ with associated $k \times 1$ vectors of parameters for each of the alternatives $\boldsymbol{\beta}_j$, so that we have $\beta_{jk} \in \{\beta_{j1}, ...\beta_{jK}\}$. After running a MNL, we'd like to know how well the model "fits" the data. To figure this out, there are several alternatives.

The "Global" Likelihood Ratio Test

This is a test vs. the global null (that is, a test for whether all coefficients $\hat{\boldsymbol{\beta}} = \mathbf{0} \,\forall\, j, k$). It is distributed $\chi^2_{(J-1)(k-1)}$ (where J is the number of possible outcomes and k is the number of covariates, *including* the constant term), and is reported automatically by most software, including Stata. It tells you if you can reject this null (which is often not very interesting or useful).

Here, the value of 951.58 indicates that we can reject the null with a very high degree of confidence. (Isn't that special?...).

Pseudo-R^2

As for the binary and ordered models, we can calculate a "pseudo"-R^2 statistic that provides a summary of model fit.

- These are usually calculated based on some function of the null and model *lnL*s.
- Stata uses one based on the ration of the log-likelihoods of the full and null models, i.e., $1 - \frac{lnL_{model}}{lnL_{null}}$.
- Maddala (1983) explains why this usually isn't a very good measure; the intuition is that, even for a "perfect fit" model, the pseudo-R^2 will be less than 1.0, and sometimes, a *lot* less...

Here, then, the pseudo-R^2 doesn't really tell us anything that we can't also get from the global LR test. Frankly, I don't advise using pseudo-R^2s in MNL and CL models, though in the end that's your choice.

Wald & Likelihood Ratio Tests on Specific Parameters/Variables

We can also do likelihood ratio (LR) and Wald tests to examine whether specific variables have effects, either singly (that is, on specific outcomes) or jointly (across all possible outcomes).

LR Tests

- *"How-To"*
  1. To do these "by hand", one just estimates models including (the *unrestricted* model) and excluding (the *restricted* model) the m variable(s) in question (with $m < k$, naturally).
  2. The LR statistic is then just (-2 $\times$ the difference of the *lnL*s).
  3. This is distributed as $\chi^2_{m(J-1)}$; that is, with $J-1$ degrees of freedom for each of the m excluded variables in the restricted model.
  4. Stata also automates this with the `-lrtest-` command; see the `help` file for more info.
- The results of this test tell you if the effect of the variable(s) is/are jointly significant across the various outcomes.
- I.e., if inclusion of those independent variable(s) helps you predict the dependent variable to a statistically significant degree. Note that this is different from individual-choice effects of the variable (i.e., t-tests).
- We can also use this method to test the hypothesis that all the variables have no joint effect on one of the outcomes (as we did above).

Wald Tests

These are asymptotically equivalent to LR tests (provided you didn't `robust`-ify your variance-covariance matrix), and can test a range of different hypotheses. The test is described Long (1997, pp. 161-2); to implement it in Stata , use the `-test-` command after estimation. The general syntax is:

```
. test [equation]variable(s)
```

where `[equation]` is the coding for the outcome you're interested in, and `variable` is/are the variable(s) you're interested in testing the effects of. Note that you need not necessarily specify the equation or variable; if you don't specify an equation, it tests the joint effects of the variable(s) indicated across all possible outcome categories, while if you don't specify a variable, it tests the joint significance of all the variables in that outcome category. We can also include equality statements, to test hypotheses about restrictions on coefficients across outcomes or variables.

So, for example, we can:

- ...test whether `age` is jointly significant:

```
. test age

 ( 1)  [2]age = 0
 ( 2)  [3]age = 0

           chi2(  2) =   10.95
         Prob > chi2 =    0.0042
```

- ...test whether *all* of the variables have a (jointly) significant effect on voting for Perot (vs. Bush):

```
. test [3]

 ( 1)  [3]partyid = 0
 ( 2)  [3]age = 0
 ( 3)  [3]white = 0
 ( 4)  [3]female = 0

           chi2(  4) =  121.13
         Prob > chi2 =    0.0000
```

- ...test whether the influence of `age` is the same on voting for Clinton (vs. Bush) as it is on voting for Perot (again, vs. Bush):

```
. test [2]age = [3]age

 ( 1)  [2]age - [3]age = 0

           chi2(  1) =    6.47
         Prob > chi2 =    0.0110
```

- ...do the same thing for the effect of `age` on voting for Bush and Perot:

```
. test [1]age = [3]age

 ( 1) - [3]age = 0

           chi2(  1) =    9.52
         Prob > chi2 =    0.0020
```

Note here that:

- Since we assume that $\hat{\beta}_{Bush} = 0$ (because he is the "baseline" category), this is the same as testing whether $(0 - \hat{\beta}_{Perot}) = 0$.
- Now, this ought to be the same as testing whether $\hat{\beta}_{Perot} = 0$ (two-tailed), right?
- And, in fact, it is: This test gives a chi-square value that's simply equal to the square of the t-test for $\hat{\beta}_{Perot} = 0$ (that is, $[-3.09]^2 = 9.52$, with some rounding error).

There are other tests that one might do; all are pretty straightforward to implement using the `-test-` command.

Aggregate Predictions and PRE

As in the binary case, we can use in-sample predictions to get a sense of how well our MNL/CL model "fits." The intuition is to generate predicted probabilities for each outcome, and then see how well the model replicates the observed distribution of outcomes on the dependent variable.

Recall that the basic probability statement for the MNL model is:

$$\Pr(Y_i = j) = \frac{\exp(\mathbf{X}_i \boldsymbol{\beta}_j)}{\sum_{j=1}^{J} \exp(\mathbf{X}_i \boldsymbol{\beta}_j)} \tag{1}$$

The MNL model therefore generates J predicted probabilities for each observation; i.e., the probability that each observation will fall into each of the J categories, as a function of its values on the independent variables $\mathbf{X}_i$. One can then classify each observation into one of the J categories, based on the highest of these probabilities, and then calculate a "reduction in error" statistic similar to that for binary logit/probit.

Using our example, we can do this "by hand":

```
. gen votehat=.
(1473 missing values generated)

. replace votehat=1 if bushhat>clinhat & bushhat>perothat
(606 real changes made)

. replace votehat=2 if clinhat>bushhat & clinhat>perothat
(829 real changes made)

. replace votehat=3 if perothat>bushhat & perothat>clinhat
(38 real changes made)

. tab2 presvote votehat, row
```

```
-> tabulation of presvote by votehat

   1=Bush, |
2=Clinton, |              votehat
   3=Perot |         1          2          3 |     Total
-----------+---------------------------------+----------
         1 |       415         77          8 |       500
           |     83.00      15.40       1.60 |    100.00
-----------+---------------------------------+----------
         2 |        56        619         16 |       691
           |      8.10      89.58       2.32 |    100.00
-----------+---------------------------------+----------
         3 |       135        133         14 |       282
           |     47.87      47.16       4.96 |    100.00
-----------+---------------------------------+----------
     Total |       606        829         38 |     1,473
           |     41.14      56.28       2.58 |    100.00
```

Note a few things:

1. A "null model" (one that chose the modal category every time) would pick all Clinton, and so would get $\left(\frac{691}{1473}\right) = 46.9\%$ correct.

2. By contrast, the estimated model predicts $\frac{(415+619+14)}{1473} = \frac{1048}{1473} = 71.15\%$ correctly.

3. So one could say this is a proportional reduction in error (PRE) of $\frac{1048-691}{1473-691} = \frac{357}{782} = 45.7\%$, in that it eliminates 45.7% of the "errors" remaining relative to a "null" model.

4. The model correctly predicts nearly 90% of the Clinton votes, and 83% of the Bush votes, but only 5% of the Perot votes. In fact, the actual Perot votes are "split" almost evenly between Clinton and Bush predictions. This is common in MNL models when some categories have very few positive outcomes overall.

At the same time, PRE statistics like this one can have their problems as well...

- There is sometimes a tendency for MNL models to predict most or all observations in one category, even if the variables seem to have significant effects; this is especially true if the dependent variable is very skewed into one category. In such cases, one's PRE won't be very impressive, since one can't improve on the "null" (modal) response very much anyway.

- One way around this problem is to calculate the predicted probabilities, then see if, for data with a particular "pattern" of independent variable values, how closely the proportion of cases in each category matches those probabilities.

Interpretation of MNL Variable Effects

Typically, interpreting MNL coefficients is not especially easy (there are lots or parameters, a nonlinear form, etc.) A few key things to remember:

1. Always bear in mind what your baseline, comparison category is; all reported results are relative to this.
2. Its generally easier to talk in terms of probabilities (either predictions or changes), or even odds ratios, than to discuss partials/first derivatives.

Odds Ratios

The MNL can be thought of as a log-odds model, where the log of the ratio of two probabilities is a function of the independent variables:

$$\ln\left[\frac{\Pr(Y_i = j|\mathbf{X})}{\Pr(Y_i = j'|\mathbf{X})}\right] = \mathbf{X}(\hat{\boldsymbol{\beta}}_j - \hat{\boldsymbol{\beta}}_{j'}) \tag{2}$$

If (as is always the case, as a practical matter) we set the coefficients of one category (say, $\hat{\boldsymbol{\beta}}_{j'}$) to zero, then we just get:

$$\ln\left[\frac{\Pr(Y_i = j|\mathbf{X})}{\Pr(Y_i = j'|\mathbf{X})}\right] = \mathbf{X}\hat{\boldsymbol{\beta}}_j$$

One nice thing about this approach is that it is *linear in the variables*; this in turn means that we can get the change in the odds ratio for category j associated with a particular variable X_k by just examining $\exp(\hat{\beta}_{jk})$...

- So for a one-unit change in X_k, the odds of observing the relevant category j (versus the baseline category) will change by $\exp(\hat{\beta}_{jk})$.
- And for a change of some value δ in X_k, the relative odds of the selected outcome j, relative to the baseline, will change by $\exp(\hat{\beta}_{jk} \times \delta)$.

Consider again the results from the example model, above. Stata will (naturally) automatically give you the odds ratios; here, they are referred to as the "relative risk ratios":

```
. mlogit, rrr

Multinomial logistic regression                   Number of obs   =       1473
                                                  LR chi2(8)      =     951.58
                                                  Prob > chi2     =     0.0000
Log likelihood = -1053.6506                       Pseudo R2       =     0.3111
------------------------------------------------------------------------------
    presvote |        RRR   Std. Err.      z    P>|z|     [95% Conf. Interval]
```

```
-------------+----------------------------------------------------------------
2            |
     partyid |   .3212245    .017623   -20.70   0.000     .2884764    .3576903
         age |   .9974021   .0051262    -0.51   0.613     .9874054      1.0075
       white |   .3719187   .1165842    -3.16   0.002     .2011984    .6874982
      female |   .8824925    .149097    -0.74   0.459     .6337247    1.228914
-------------+----------------------------------------------------------------
3            |
     partyid |   .6057295   .0294976   -10.29   0.000     .5505886    .6663927
         age |   .9845555   .0049657    -3.09   0.002     .9748708    .9943365
       white |   2.408925   1.050425     2.02   0.044     1.024831    5.662322
      female |   .6009293    .097748    -3.13   0.002     .4368836    .8265726
------------------------------------------------------------------------------
(presvote==1 is the base outcome)
```

These tell us that:

- A one unit increase in `partyid` corresponds to:
  - A decrease in the log-odds of a Clinton vote, versus a vote for Bush, of $\exp(-1.136) = 0.321$ (or about 68 percent), and
  - A decrease in the log-odds of a Perot vote, versus a vote for Bush, of $\exp(-0.501) = 0.606$ (or about 40 percent).
  - These are *large* decreases in the odds – not surprisingly, more Republican voters are *much* more likely to vote for Bush than for Perot or Clinton.
- Similarly, female voters are:
  - No more or less likely to vote for Clinton than for Bush, but
  - Roughly 40 percent less likely to have voted for Perot.
- And so forth...

Note that `Stata` also reports the standard errors and confidence intervals for the relative risk ratios, which are useful as well. If you're interested in other comparisons (e.g., the effect of changes in `partyid` on voting for Perot versus Clinton), it is probably easiest just to rerun the model with a different "baseline" category to get the relevant statistics.

Partial Changes / Marginal Effects

The partial change in $\Pr(Y_i = j)$ for a particular variable X_k is:

$$\frac{\partial \Pr(Y_i = j)}{\partial X_k} = \Pr(Y_i = j|\mathbf{X}) \left[\hat{\boldsymbol{\beta}}_{jk} - \sum_{j=1}^{J} \hat{\boldsymbol{\beta}}_{jk} \times \Pr(Y_i = j|\mathbf{X}) \right] \tag{3}$$

Note a few things about this:

- The marginal effect varies as a function of a bunch of things, including
  - The probability itself,
  - The value of the coefficient estimate,
  - The sums of the other coefficients for that covariate.
- This means that the $\left[\hat{\boldsymbol{\beta}}_{jk} - \sum_{j=1}^{J} \hat{\boldsymbol{\beta}}_{jk} \times \Pr(Y_i = j|\mathbf{X})\right]$ term signs the marginal effect, which in turn means that the marginal effect may or may not have the same sign as the coefficient estimate itself.

You will be unsurprised to learn that `Stata` will calculate these for you, using the `-mfx-` command. Note that one has to do so separately for each of the J possible outcomes:

```
. mfx, predict(p outcome(1))

Marginal effects after mlogit
      y  = Pr(presvote==1) (predict, p outcome(1))
         =  .28317841
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
 partyid |   .1847824      .00899   20.55   0.000   .167161  .202404   3.75492
     age |   .0014627      .00092    1.60   0.110  -.000332  .003258   45.8873
   white*|      .1016      .04799    2.12   0.034   .007532  .195668   .878479
  female*|   .0529949      .02972    1.78   0.075  -.005254  .111243   .514596
------------------------------------------------------------------------------
(*) dy/dx is for discrete change of dummy variable from 0 to 1


. mfx, predict(p outcome(2))

Marginal effects after mlogit
      y  = Pr(presvote==2) (predict, p outcome(2))
         =  .46220238
```

```
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
 partyid |   -.223283      .01092  -20.45   0.000  -.244678 -.201888   3.75492
     age |   .0011852      .00111    1.07   0.284  -.000984  .003355   45.8873
   white*|  -.3135228      .05406   -5.80   0.000  -.419479 -.207567   .878479
  female*|   .0290831      .03582    0.81   0.417  -.041126  .099292   .514596
------------------------------------------------------------------------------
(*) dy/dx is for discrete change of dummy variable from 0 to 1


. mfx, predict(p outcome(3))

Marginal effects after mlogit
      y  = Pr(presvote==3) (predict, p outcome(3))
         =  .25461921
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
 partyid |   .0385005      .00766    5.03   0.000    .02349  .053511   3.75492
     age |  -.0026479      .00083   -3.20   0.001   -.00427 -.001026   45.8873
   white*|   .2119229      .03084    6.87   0.000   .151475  .272371   .878479
  female*|  -.0820779      .02659   -3.09   0.002  -.134191 -.029965   .514596
------------------------------------------------------------------------------
(*) dy/dx is for discrete change of dummy variable from 0 to 1
```

Notice here that:

- In its usual way, Stata reports discrete probability changes for dummy covariates; the others can be thought of as "slopes."
- Also, many of the marginal effects have different signs than the coefficients themselves. For example, the $\hat{\beta}$ for `partyid` for Perot (vs. Bush) is strongly negative, but the marginal effect (at means of the covariates) is positive. We'll see an illustration of why this is the case in a bit, when we get to predicted probabilities.

For several reasons (outlined at some length in Long and Maddala), the partial effect is not the best of a variable's influence on $\Pr(Y_i = j)$...

- It depends (almost) entirely on the values at which the other variables are set when the derivative is taken,
- It may or may not have the same sign as the coefficient itself, and

- Its sign may change with the value of the variable in question (not unlike in the ordered models...).

In my humble opinion, in general, you're better off not using marginal effects in MNL/CL models; there are better interpretation methods out there.

Predicted Probabilities and Probability Changes

We can use the basic probability statement of the MNL/CL model to generate predictions for the probability of each category, a lá binary logit/probit. As in those models, we're required to select and set the values of the other independent variables (typically means or medians). We can then do the usual stuff:

- Examine predictions across ranges of independent variables.
- Examine changes in predictions with unit/std. dev./min-max changes in independent variables.
- Plot any/all of the above, as well as their confidence intervals.

In-Sample Predictions

The `-predict-` command in Stata will generate J in-sample predicted values for each observation, one for each category of the dependent variable. However, when using `-predict-`, it is crucial to tell Stata which equation (outcome) you want the predicted values for, using the `-outcome-` option:

```
. predict bushhat, outcome(1)
(option pr assumed; predicted probability)
```

This creates a new variable `bushhat`, which is equal to the model-predicted probability that each voter in the sample would vote for Bush, based on the values of his or her covariates; that is,

$$\begin{aligned}\Pr(\widehat{\texttt{presvote}_i} = \text{Bush}) &= \frac{\exp(\mathbf{X}_i\hat{\boldsymbol{\beta}}_{\text{Bush}})}{\sum_{j=1}^{J}\exp(\mathbf{X}_i\hat{\boldsymbol{\beta}}_j)} \\ &= \frac{1}{\sum_{j=1}^{J}\exp(\mathbf{X}_i\hat{\boldsymbol{\beta}}_j)}\end{aligned}$$

Similarly:

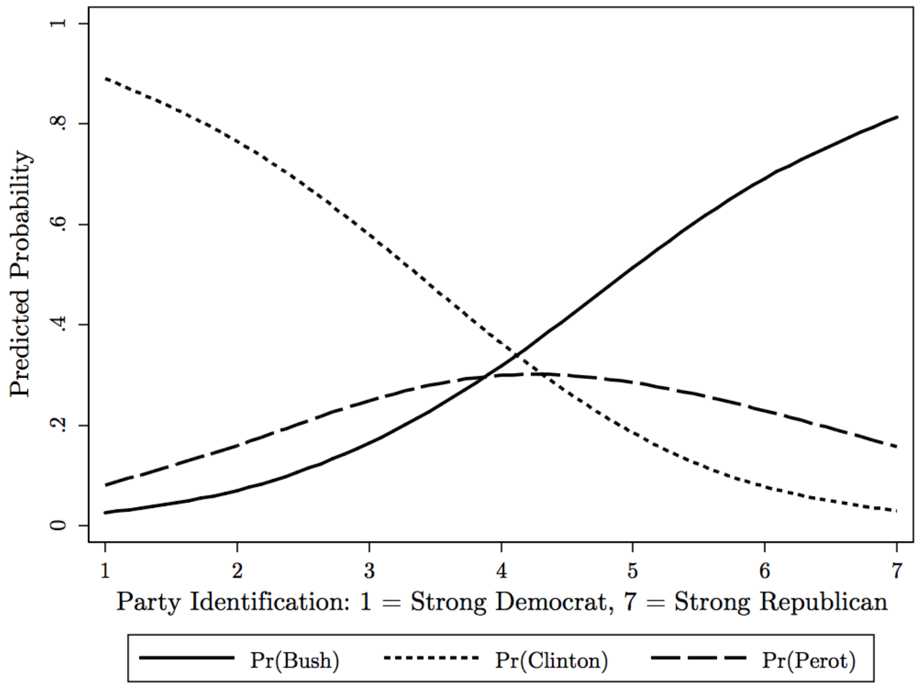
```
. predict clinhat, outcome(2)
(option pr assumed; predicted probability)

. predict perothat, outcome(3)
(option pr assumed; predicted probability)
```

yield predictions for Clinton and Perot, respectively.

What do we do with these? One possibility is to plot these predicted probabilities as a function of one (or more) covariate(s). Here, we'll again focus in `partyid`:

```
. twoway (mspline bushhat partyid, sort lcolor(black) lpattern(solid) lwidth(medthick))
(mspline clinhat partyid, sort lcolor(black) lpattern(shortdash) lwidth(medthick))
(mspline perothat partyid, sort lcolor(black) lpattern(longdash) lwidth(medthick)),
ytitle(Predicted Probability) xtitle("Party Identification: 1 = Strong Democrat, 7
= Strong Republican") xscale(range(1. 7.)) xlabel(1(1)7) legend(cols(3) order(1
"Pr(Bush)" 2 "Pr(Clinton)" 3 "Pr(Perot)"))
```

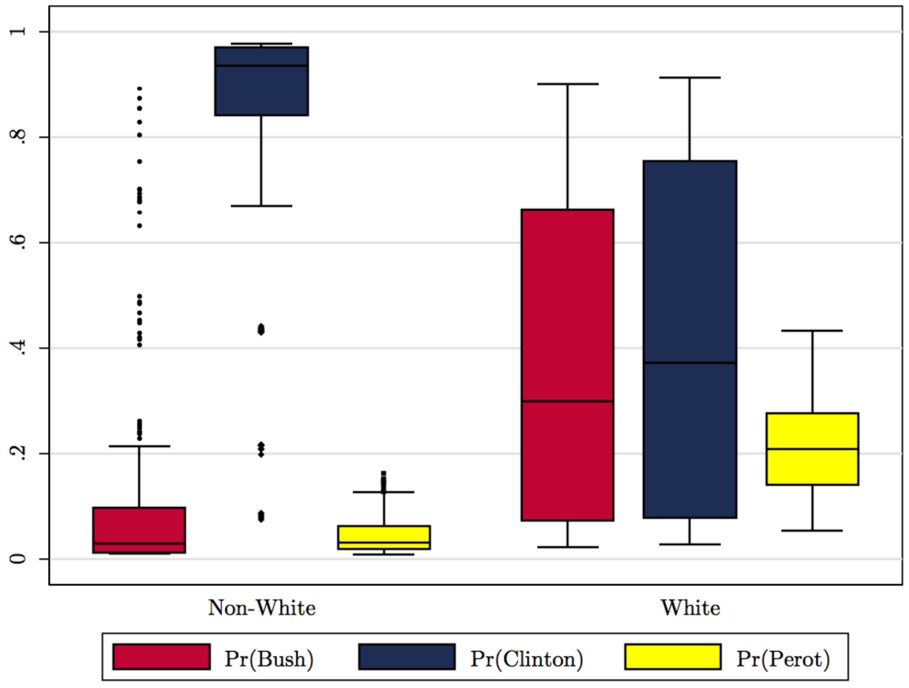
Figure 1: In-Sample Predicted Probabilities, by `partyid` (Median Splines)



The results conform with what we might expect: Democrats voting for Clinton, Republicans for Bush, and (mostly) independents for Perot. One can do a similar thing for binary covariates, in which case something like a boxplot is often better:

```
. graph box bushhat clinhat perothat, medtype(cline) medline(lcolor(black) lpattern(solid)
lwidth(medium)) over(white) box(1, fcolor(cranberry) lcolor(black)) box(2, fcolor(dknavy)
lcolor(black)) box(3, fcolor(yellow) lcolor(black)) marker(1, msymbol(smcircle) msize(vsmall)
mcolor(black)) marker(2, msymbol(smdiamond) msize(vsmall) mcolor(black)) marker(3,
msymbol(smsquare) msize(vsmall) mcolor(black)) legend(cols(3) order(1 "Pr(Bush)" 2
"Pr(Clinton)" 3 "Pr(Perot)"))
```

Figure 2: In-Sample Predicted Probabilities, by `white` (Boxplots)



This latter plots shows the median and distribution of voters' predicted probabilities of voting for each candidate, across different values of `white`. Note that there are large differences between whites and non-whites on the probabilities for Bush and Clinton, and relatively smaller ones for Perot.

Out-of-Sample Predictions

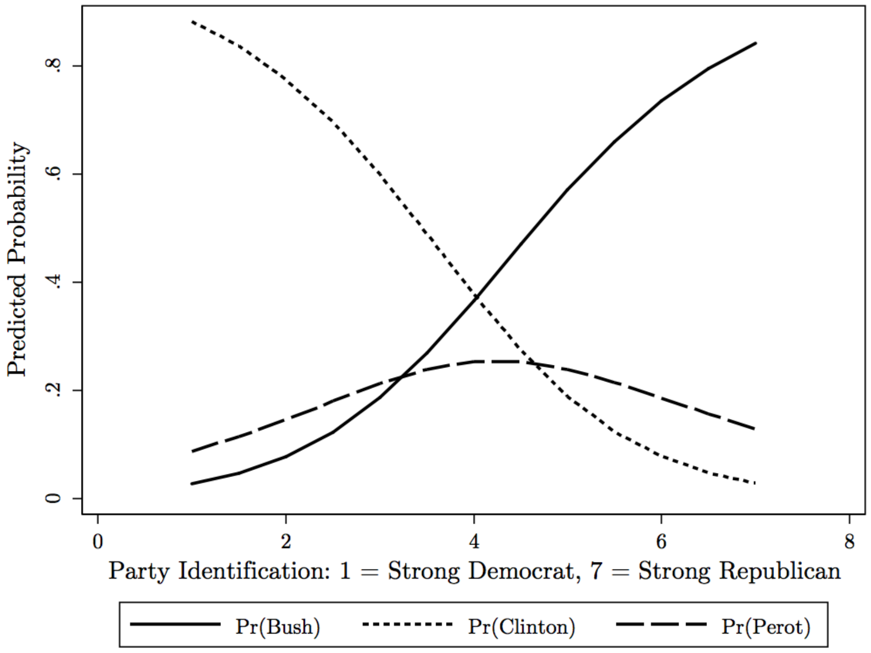
In a manner analogous to that for binary response models, we can also generate out-of-sample predictions on simulated data, in order to illustrate hypothetical examples. Here, we'll focus on changes in the predicted probabilities across different values of `partyid`. Doing so is accomplished exactly like before:

```
. clear

. set obs 13
obs was 0, now 13

. gen partyid = (_n+1)/2

. gen age=45.887

. gen white=1
. gen female=1

. save MNLsim.dta
file MNLsim.dta saved

. use MNLData.dta

. mlogit presvote partyid age white female, basecategory(1)

(output omitted)

. use MNLsim.dta

. predict bushhat, outcome(1)
(option pr assumed; predicted probability)

. predict clinhat, outcome(2)
(option pr assumed; predicted probability)

. predict perothat, outcome(3)
(option pr assumed; predicted probability)

. gen zero=0

. gen one=1

. gen bushperot=bushhat+perothat
```

These last three variables will come in handy in just a bit. The natural first thing to do is to plot the predicted probabilities as a function of the variable of interest:

```
. twoway (line bushhat partyid, sort lcolor(black) lpattern(solid) lwidth(medthick))
(line clinhat partyid, sort lcolor(black) lpattern(shortdash) lwidth(medthick)) (line
perothat partyid, sort lcolor(black) lpattern(longdash) lwidth(medthick)), ytitle(Predicted
Probability) xtitle("Party Identification: 1 = Strong Democrat, 7 = Strong Republican")
legend(cols(3) order(1 "Pr(Bush)" 2 "Pr(Clinton)" 3 "Pr(Perot)"))
```

Figure 3: Out-Of-Sample Predicted Probabilities, by `partyid`

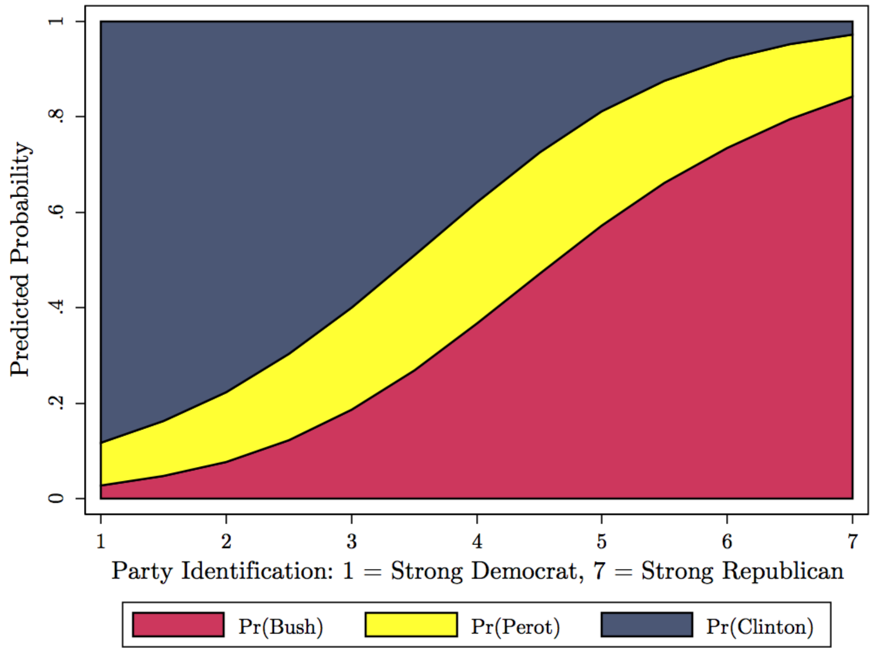


Not surprisingly, this looks a lot like the in-sample smoothed plot: Democrats support Clinton, Republicans Bush, and independents Perot.

One can also plot something akin to "cumulative" probabilities across different categories:

```
. twoway (rarea bushhat zero partyid, sort lcolor(black) fcolor(cranberry)) (rarea
bushhat bushperot partyid, sort lcolor(black) fcolor(yellow)) (rarea bushperot one
partyid, sort lcolor(black) fcolor(dknavy)), ytitle(Predicted Probability) xtitle("Party
Identification: 1 = Strong Democrat, 7 = Strong Republican") xscale(range(1. 7.))
xlabel(1(1)7) legend(cols(3) order(1 "Pr(Bush)" 2 "Pr(Perot)" 3 "Pr(Clinton)"))
```

Figure 4: “Cumulative” Out-Of-Sample Predicted Probabilities, by `partyid`



Here, the “thickness” of the region at any given value of `partyid` tells you the (median predicted) probability that a voter with that level of party identification will choose that candidate.

Finally, we can use `-predict-` to generate confidence intervals around these predictions, though this takes just a little extra work:

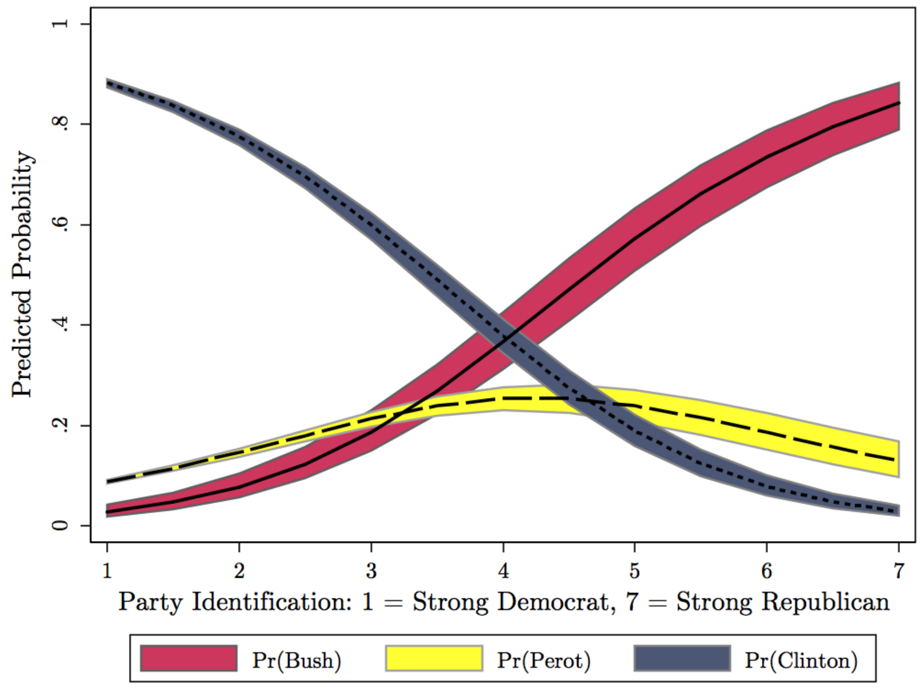
```
. predict bushXB, xb outcome(1)

. predict clintonXB, xb outcome(2)

. predict perotXB, xb outcome(3)

. predict bushSE, stdp outcome(1)

. predict clintonSE, stdp outcome(2)

. predict perotSE, stdp outcome(3)

. gen bushUB = (exp(bushXB+(1.96*bushSE))) / ((exp(bushXB+(1.96*bushSE)) +
(exp(clintonXB+(1.96*clintonSE))) + (exp(perotXB+(1.96*perotSE))))

. gen bushLB = (exp(bushXB-(1.96*bushSE))) / ((exp(bushXB-(1.96*bushSE)) +
(exp(clintonXB-(1.96*clintonSE))) + (exp(perotXB-(1.96*perotSE))))
```

```
. gen clintonUB = (exp(clintonXB+(1.96*clintonSE))) / ((exp(bushXB+(1.96*bushSE))) +
(exp(clintonXB+(1.96*clintonSE))) + (exp(perotXB+(1.96*perotSE))))

. gen clintonLB = (exp(clintonXB-(1.96*clintonSE))) / ((exp(bushXB-(1.96*bushSE))) +
(exp(clintonXB-(1.96*clintonSE))) + (exp(perotXB-(1.96*perotSE))))

. gen perotUB = (exp(perotXB+(1.96*perotSE))) / ((exp(bushXB+(1.96*bushSE))) +
(exp(clintonXB+(1.96*clintonSE))) + (exp(perotXB+(1.96*perotSE))))

. gen perotLB = (exp(perotXB-(1.96*perotSE))) / ((exp(bushXB-(1.96*bushSE))) +
(exp(clintonXB-(1.96*clintonSE))) + (exp(perotXB-(1.96*perotSE))))
```

Which we can then plot as:

```
. twoway (rarea bushUB bushLB partyid, sort fcolor(cranberry)) (rarea perotUB perotLB
partyid, sort fcolor(yellow)) (rarea clintonUB clintonLB partyid, sort fcolor(dknavy)) (line
bushhat partyid, lcolor(black) lpattern(solid) lwidth(medthick)) (line clinhat partyid,
lcolor(black) lpattern(shortdash) lwidth(medthick)) (line perothat partyid, lcolor(black)
lpattern(longdash) lwidth(medthick)), ytitle(Predicted Probability) xtitle("Party Identification:
1 = Strong Democrat, 7 = Strong Republican") xscale(range(1. 7.)) xlabel(1(1)7)
legend(cols(3) order(1 "Pr(Bush)" 2 "Pr(Perot)" 3 "Pr(Clinton)"))
```

Figure 5: Out-Of-Sample Predicted Probabilities, by `partyid`, with 95% Confidence Intervals



Other Stata - Based Alternatives

Finally, there are the usual other tools for calculating differences in predicted values associated with changes in the independent variables.

- Long's -spost- package of routines in Stata will automatically calculate predicted values and changes in predicted values associated with changes in covariates; see the routines for more details.
- In addition, Clarify also works with the -mlogit- command.

Conditional Logit

Interpreting conditional logit results is actually somewhat easier than those for MNL, in part because the response variable is a binary indicator, just like in standard binary logit/probit models. The example model we'll work with is this one, from the 1992 presidential election:

```
. use CLdata.dta, clear

. clogit vote FT clintondummy perotdummy PIDxClinton PIDxPerot, group(caseid)

Conditional (fixed-effects) logistic regression   Number of obs   =       4419
                                                  LR chi2(5)      =    1764.49
                                                  Prob > chi2     =     0.0000
Log likelihood =  -736.0092                       Pseudo R2       =     0.5452
------------------------------------------------------------------------------
        vote |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          FT |   .0629875   .0032175    19.58   0.000     .0566813    .0692937
clintondummy |   2.812737   .2687999    10.46   0.000     2.285899    3.339576
  perotdummy |   .9435437   .2856252     3.30   0.001     .3837286    1.503359
 PIDxClinton |  -.6318723    .062255   -10.15   0.000    -.7538899   -.5098548
   PIDxPerot |  -.1921175    .057032    -3.37   0.001    -.3038981   -.0803369
------------------------------------------------------------------------------
```

The model thus incorporates both covariates that vary only across respondents (partyid) and one that varies across both alternatives and respondents (the "feeling thermometer" variable FT).

Odds Ratios and Marginal Effects

The CL model is also a model of proportional odds, which means that the odds ratio has a particularly simple form:

$$\ln\left[\frac{\widehat{\Pr(Y_{ij}=1}|\mathbf{X})}{\widehat{\Pr(Y_{ij}=0}|\mathbf{X})}\right] = \mathbf{X}\hat{\boldsymbol{\beta}} \tag{4}$$

which in turn means that the odds ratio associated with a one-unit change in some particular covariate X_k is just

$$\widehat{OR_{\Delta X_k = 1}} = \exp(\hat{\beta}_k)$$

Of course, we can generate these things automatically, using the `-or-` subcommand:

```
. clogit, or

Conditional (fixed-effects) logistic regression   Number of obs   =       4419
                                                  LR chi2(5)      =    1764.49
                                                  Prob > chi2     =     0.0000
Log likelihood =  -736.0092                       Pseudo R2       =     0.5452
------------------------------------------------------------------------------
        vote | Odds Ratio   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
          FT |   1.065014   .0034267    19.58   0.000     1.058318    1.071751
clintondummy |   16.65545   4.476983    10.46   0.000     9.834527    28.20715
  perotdummy |   2.569069   .7337909     3.30   0.001     1.467747    4.496767
 PIDxClinton |   .5315956   .0330945   -10.15   0.000     .4705327    .6005828
   PIDxPerot |   .8252099   .0470633    -3.37   0.001     .7379361    .9228054
------------------------------------------------------------------------------
```

This lets us know that (e.g.) a one-degree increase in the feeling thermometer towards a particular candidate increases the odds of voting for that candidate by about 6.5 percent, with a 95% c.i. of 5.8 - 7.2 percent. The interpretation is a bit more challenging for the observation-specific variables, since they require that we consider their interaction with the choice-specific dummy variables, but doing so only requires the same straightforward application of techniques for interpreting interaction terms we've used before.

We can also consider the marginal effects (partial derivatives) for the conditional logit model. Those have a somewhat more straightforward interpretation than in the MNL, though they still depend critically on the existing probabilities. The `-mfx-` command in Stata once again calculates these automatically:

```
. mfx, predict(pu0)

Marginal effects after clogit
      y  = Pr(vote|fixed effect is 0) (predict, pu0)
         =  .96831689
------------------------------------------------------------------------------
variable |      dy/dx    Std. Err.     z    P>|z|  [    95% C.I.   ]      X
---------+--------------------------------------------------------------------
      FT |   .0019324      .00024    8.03   0.000   .001461  .002404   50.7882
clinto~y*|   .0721238      .01344    5.36   0.000   .045775  .098472   .333333
perotd~y*|   .0257466      .00749    3.44   0.001   .011061  .040432   .333333
PIDxCl~n | -.0193854      .00392   -4.94   0.000  -.027077 -.011694   1.25164
PIDxPe~t |  -.005894      .00196   -3.01   0.003  -.009726 -.002062   1.25164
------------------------------------------------------------------------------
(*) dy/dx is for discrete change of dummy variable from 0 to 1
```

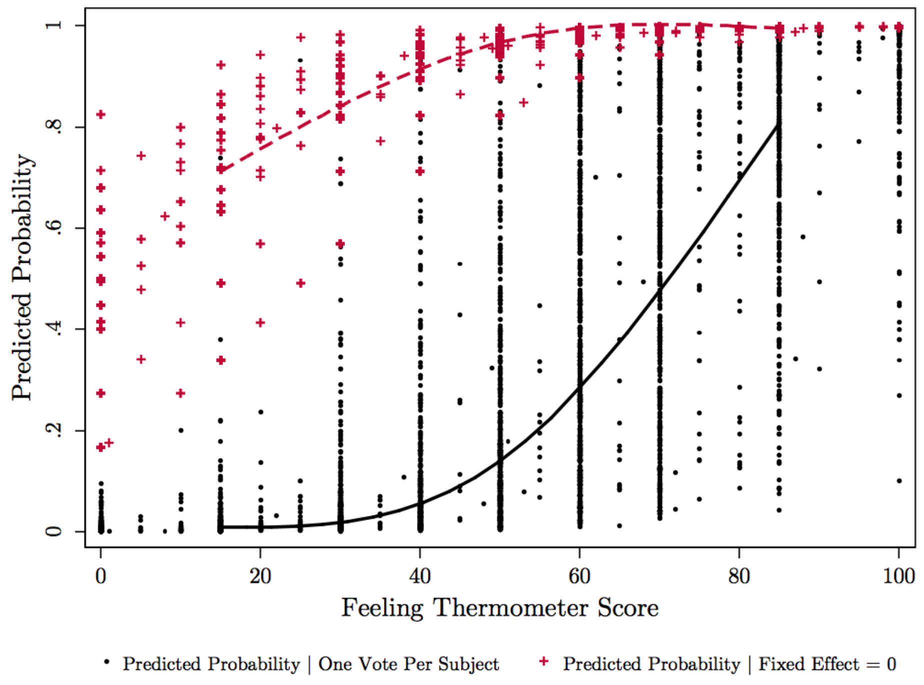
Importantly, here we have to specify the `-pu0-` prediction option; the results are marginal effects calculated assuming that the observation-specific "fixed effect" is equal to zero (more on this below). In this case, that means that the "baseline" probability is quite high, which means that the "slopes" / changes are very small; one could use the `-at()-` option in `-mfx-` to change this.

Predicted Probabilities

Finally, we can (and should) calculate and plot predicted probabilities, both in- and out-of-sample, as well as their associated measures of uncertainty. Here again, the approach is familiar; for the in-sample variety, for example, we use:

```
. predict votehat
(option pc1 assumed; conditional probability for single outcome within group)

. predict altvotehat, pu0

. twoway (scatter votehat FT, sort msymbol(smcircle) mcolor(black) msize(vsmall)) (scatter
altvotehat FT, sort msymbol(plus) mcolor(cranberry) msize(medsmall)) (mspline votehat
FT, sort bands(3) lcolor(black) lpattern(solid) lwidth(medthick)) (mspline altvotehat FT,
sort bands(3) lcolor(cranberry) lpattern(dash) lwidth(medthick)), ytitle(Predicted Probability)
xtitle(Feeling Thermometer Score) xscale(range(0. 100.)) xlabel(0(20)100) legend(order(1
"Predicted Probability" 2 "Predicted Probability | Fixed Effect = 0") size(medsmall))
legend(region(lcolor(none)))
```

Figure 6: In-Sample Predicted Probabilities, by `FT` (with median splines)



Here, I've left the actual (noisy) predictions in the plot as well as their smoothed values. Notice the differences:

- The circles are the default ("`pc1`") predictions; these assume that there can be only one positive response per subject, and so restrict the predictions so that $\sum_{j=1}^{J} \widehat{\Pr(Y_{ij} = 1|\mathbf{X}_i)} = 1.0$. In-sample, these are almost always the one we want to pay attention to.
- The crosses are the predictions assuming that the observation-level "fixed effect" – that is, the subject-specific intercept that is part of the CL model – is equal to zero (denoted by "`pu0`"). As a result,
  - The predictions within an observation no longer need sum to 1.0, and
  - On average, these predicted values are larger than those for `pc1`.

One could do a similar set of plots for out-of-sample predictions, using simulated data and following the general contours described for the MNL model (above).

Finally: MNL / CL are far from the only model for unordered, categorical dependent variables. Next time, we'll discuss some others, and do some comparisons.