Discrete Choice Models II

1 Introduction

Last time we examined several commonly used discrete choice models: conditional logit, multinomial logit, and mixed conditional logit.[1] This time, we will look at slightly more advanced discrete choice models: nested logit, multinomial probit, and mixed logit. Although these models are less common, they do open up a lot of interesting avenues of research.

2 Generalized Extreme Values Models – Nested Logit

Generalized Extreme Value (GEV) models allow for a variety of substitution patterns among the alternatives. The common attribute among all GEV models is that they assume that the error terms are distributed according to a generalized extreme value distribution. GEV models, as the name suggests, are a generalization of the univariate extreme value distribution that is used in the MNL and CL models examined earlier. When all the correlations in a GEV model are 0, the GEV model becomes the product of independent extreme value distributions and the GEV model becomes a standard logit model. You can test to see if the correlations are 0, thereby testing whether the standard logit model is an accurate reflection of the substitution patterns. The most widely used member of the GEV family of models is the nested logit (NL) model.

Train (2007, 81) notes that the nested logit model is appropriate when the choice set facing a decision maker can be partitioned into subsets, known as nests, in such a way that the following properties hold:

1. For any two alternatives in the same nest, the ratio of probabilities is independent of the attributes or existence of all other alternatives in the nest. In other words, IIA holds within each nest.
2. For any two alternatives in different nests, the ratio of probabilities can depend on the attributes of other alternatives in the two nests. In other words, IIA does not hold in general for alternatives in different nests.

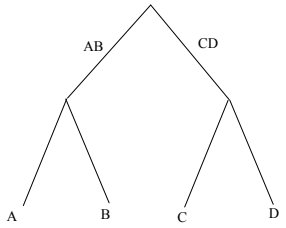
Train provides the following example. The choice set includes four alternatives: A, B, C, and D. As we saw earlier, if any alternative is removed, the probabilities of the other alternatives will necessarily rise. But in what proportion do the probabilities rise? The thing to notice is that the probabilities of C and D always rise by the same proportion when one of the other alternatives is removed. We could therefore put these two alternatives in their own 'CD' nest because IIA holds between them. Similarly, the probabilities of A and B also always rise by the same proportion when one of the other alternatives is removed. We could therefore put these two alternatives in their own 'AB' nest as well because IIA holds between them. As Table 1 indicates, IIA does not hold between choices (C, D) and (A, B). A convenient way to capture the substitution patterns in this example is with the help of a decision tree like the one in Figure 1.

[1] These notes are heavily based on Train's (2007) excellent book on discrete choice models.

Table 1: Example of IIA holding within Nests

Alternative	Original	A	B	C	D
A	0.40	·	0.45 (+12.5%)	0.52 (+30%)	0.48 (+20%)
B	0.10	0.20 (+100%)	·	0.13 (+30%)	0.12 (+20%)
C	0.30	0.48 (+60%)	0.33 (+10%)	·	0.40 (+33%)
D	0.20	0.32 (+60%)	0.22 (+10%)	0.35 (+70%)	·

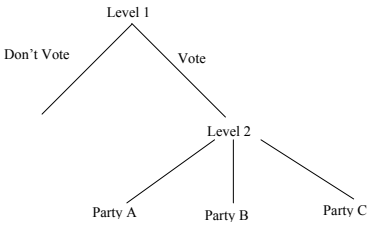
Figure 1: An Example of a Nested Logit Model with 2 Levels



As you can see, nested logit models are often appropriate when we think that our choice problem has two or more levels, or that there is a sequential nature to the choice problem. Note, though, that although decision trees in nested logit analyses, like the one shown in Figure 1, are often interpreted as implying that the highest-level decisions are made first, followed by decisions at lower levels, no such temporal ordering is necessarily implied (Henscher, Rose & Greene 2005, Chapter 13). In fact, probably the best way to think about nested logit models is that they are appropriate when we believe that we have groups of alternatives that are similar to each other (other alternatives in the group or nest) in unobserved ways; in other words, they are appropriate when there is correlation for unobserved reasons between the alternatives in each nest but no correlation between alternatives in different nests.

As another example of a situation in which a nested logit model might be appropriate, we might think that individuals first decide whether to vote or not, and then if they decide to vote, then they decide whether to vote for Party A, B, C etc. This would be another nested logit model with two levels.

Figure 2: Nested Logit Model with 2 Levels



In terms of the Dutch elections data that we have been using, we could think that an individual first chooses

whether to vote for a secular party (alternatives 1, 3, 4) or a religious party (2). If the individual chooses to vote for a secular party, then he decides whether to vote for alternative 1, 3 or 4.

2.1 Choice Probabilities

One way to come up with the nested logit model is the following. Let the set of alternatives j be partitioned into K nonoverlapping subsets denoted $B_1, B_2, \ldots B_K$ and called nests. The utility that individual n obtains from alternative j in nest B_k is denoted in the usual manner as $U_{nj} = V_{nj} + \epsilon_{nj}$. The nested logit model is obtained by assuming that the vector of unobserved utility, $\epsilon_n = \langle \epsilon_{n1}, \epsilon_{n2}, \ldots, \epsilon_{nJ} \rangle$ has the following cumulative distribution:

$$\exp\left(-\sum_{k=1}^{K}\left(\sum_{j\in B_k} e^{-\epsilon_{nj}/\lambda_k}\right)^{\lambda_k}\right) \tag{1}$$

This distribution is a type of GEV distribution. For a logit model, each ϵ_{nj} is independent with a univariate extreme value distribution. However, the ϵ_{nj}'s are correlated within the nests. For any two alternatives j and m in nest B_k, ϵ_{nj} is correlated with ϵ_{nm}. For any two alternatives in different nests, the unobserved portion of utility is still uncorrelated: $\text{cov}(\epsilon_{nj}, \epsilon_{nm}) = 0$ for any $j \in B_k$ and $m \in B_\iota$ with $\iota \neq k$.

The parameter λ_k is a measure of the degree of independence in unobserved utility among the alternatives in nest B_k; it is sometimes referred to as a dissimilarity parameter. A high λ_k means greater independence and less correlation i.e. the alternatives in the nest are less similar for unobserved reasons. The statistic $1 - \lambda_k$ provides a measure of correlation i.e. when this statistic is high, there is more correlation and when this statistic is low, there is less correlation. A value of $\lambda_k = 1$ means complete independence in nest B_k. Obviously, if $\lambda_k = 1$ for all nests, then the GEV distribution simply becomes the produce of independent extreme value terms i.e. the nested logit reduces to the standard logit model.

With this distribution, the probability that individual n chooses alternative i from the choice set is:

$$P_{ni} = \frac{e^{V_{ni}/\lambda_k}\left(\sum_{j\in B_k} e^{V_{nj}/\lambda_k}\right)^{\lambda_k - 1}}{\sum_{\iota=1}^{K}\left(\sum_{j\in B_\iota} e^{V_{nj}/\lambda_\iota}\right)^{\lambda_\iota}} \tag{2}$$

From this equation it is relatively easy to show that IIA holds within nests but not across nests. Consider alternatives $i \in B_k$ and$m \in B_\iota$. Since the denominator in Eq. (3) is the same for all alternatives, the ratio of probabilities for these two alternatives is just:

$$\frac{P_{ni}}{P_{nm}} = \frac{e^{V_{ni}/\lambda_k}\left(\sum_{j\in B_k} e^{V_{nj}/\lambda_k}\right)^{\lambda_k - 1}}{e^{V_{nm}/\lambda_\iota}\left(\sum_{j\in B_\iota} e^{V_{nj}/\lambda_\iota}\right)^{\lambda_\iota - 1}} \tag{3}$$

If $k = \iota$, or alternatively i and m are in the same nest, then the stuff in parentheses cancel out and we get:

$$\frac{P_{ni}}{P_{nm}} = \frac{e^{V_{ni}/\lambda_k}}{e^{V_{nm}/\lambda_\iota}} \tag{4}$$

As you can see, this last ratio is independent of all other alternatives i.e. we have IIA within nests. In contrast, if $k \neq \iota$, or alternatively i and m are in different nests, then the stuff in parentheses does not cancel out and so the ratio of probabilities depends on the attributes of all alternatives in the nests that contain i and m. Note, though, that in this latter case, the probabilities still do not depend on the attributes of alternatives that are not in the nests containing i and m. In other words, we have what Ken Train calls the independence from irrelevant nests (IIN). Thus, in a nested logit model we do have a relaxation of the IIA assumption compared to a normal logit model but you still have (i) IIA holding over alternatives in each nest and (ii) IIN holding over alternatives in different nests.

Another thing to note is that λ_k is subscripted by k. In other words, the parameter λ_k can differ over nests, reflecting different correlation among unobserved factors within each nest. It is possible for the analyst to constrain the λ_k's to be the same for all (or some) of the nests, indicating that the degree of correlation is the same in each of these nests. You can conduct hypothesis tests to see if these constraints are reasonable using a likelihood ratio test.

Another thing worth noting is that λ_k must be within a particular range if the model is to be consistent with utility-maximizing behavior. Specifically, $\lambda_k \; \forall k$ must be between 0 and 1. If $\lambda_k > 1$, then the model is consistent with utility-maximizing behavior for only some range of the independent variables. A negative value of λ_k is inconsistent with utility maximization since it implies that improving the attributes of an alternative actually decreases the probability that it will be chosen. One way to think about this is that an estimated λ_k outside the (0, 1] bounds suggests a misspecification problem with the model: the systematic component could be misspecified, the grouping could be misspecified, or both could be misspecified.

2.1.1 An Alternative Presentation

Most textbooks do not present the choice probability from a nested logit model in the way shown in Eq. (3). It turns out that there is an alternative and more intuitive way of presenting it. It is possible to decompose the observed portion of utility into two parts: (i) a part labeled W that is constant for all alternatives within a nest and (ii) a part labeled Y that varies over alternatives within a nest. Thus, you have

$$U_{nj} = W_{nk} + Y_{nj} + \epsilon_{nj} \tag{5}$$

for $j \in B_k$, where

- W_{nk} depends only on variables that describe nest k. These variables differ over nests but not over alternatives within each nest.
- Y_{nj} depends on variables that describe alternative j. These variables vary over alternatives within nest k.

Decomposing the observed portion of utility in this way, allows us to write the nested logit probability as the product of two standard logit probabilities. In effect, we can let the probability of choosing alternative $i \in B_k$ be equal to the probability that nest B_k is chosen multiplied by the probability that alternative i is chosen given that an alternative in B_k is chosen:

$$P_{ni} = P_{nB_k} \times P_{ni|B_k} \tag{6}$$

where $P_{ni|B_k}$ is the conditional probability of choosing i given that an alternative in nest B_k is chosen, and P_{nB_k} is the marginal probability of choosing an alternative in nest B_k. These conditional and marginal probabilities take the form of logits and so can be written as follows:[2]

$$P_{nb_k} = \frac{e^{W_{nk}+\lambda_k I_{nk}}}{\sum_{\iota=1}^{K} e^{W_{n\iota}+\lambda_\iota I_{n\iota}}} \tag{7}$$

$$P_{ni|b_k} = \frac{e^{Y_{ni}/\lambda_k}}{\sum_{j\in B_k} e^{Y_{nj}/\lambda_k}} \tag{8}$$

where

$$I_{nk} = \ln \sum_{j\in B_k} e^{Y_{nj}/\lambda_k} \tag{9}$$

Thus, we have:

$$P_{ni} = \frac{e^{W_{nk}+\lambda_k I_{nk}}}{\sum_{\iota=1}^{K} e^{W_{n\iota}+\lambda_\iota I_{n\iota}}} \times \frac{e^{Y_{ni}/\lambda_k}}{\sum_{j\in B_k} e^{Y_{nj}/\lambda_k}} \tag{10}$$

It is perhaps easier to see the intuition here by thinking of the two-levels in the nested logit that we are examining. The probability in Eq. (10) essentially tells us the probability that alternative k (nest B_k) is chosen in the first level and alternative i is chosen in the second level. Let $W_{nk} = w_{nk}\gamma$ and $Y_{nj} = x_{nj}\beta$. Substituting in we have:

$$P_{nB_k i} = \frac{e^{w_{nk}\gamma+\lambda_k I_{nk}}}{\sum_{\iota=1}^{K} e^{w_{n\iota}\gamma+\lambda_\iota I_{n\iota}}} \times \frac{e^{x_{ni}\beta/\lambda_k}}{\sum_{j\in B_k} e^{x_{nj}\beta/\lambda_k}} \tag{11}$$

where $P_{nB_k i}$ is the probability that individual n chooses nest B_k in the first level and alternative i in the second level, K is the number of discrete choices or nests in the first level, w_{nk} is a matrix of independent variables associated with nest B_k in the first level, x_{ni} is a matrix of independent variables associated with alternative i in the second level, and γ, λ, and β are vectors of coefficients.[3]

[2] To see why, see Train (2007, 90).

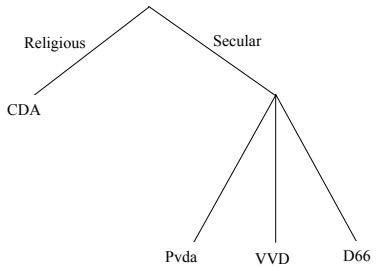
[3] Note that you will sometimes see the nested logit choice probability written without the coefficients in the second level being divided by λ_k. For example, this is what happens in Greene (2003). If the coefficients in the lower model are not divided by λ_k, then the choice probability is not the same as that shown in Eq. (10). The model that I show in which the coefficients in the bottom level are dividing by λ_k is sometimes referred to as the normalized nested logit model and it is consistent with a random utility model (RUM) setup. By scaling the coefficients within each nest, the RUM-consistent model allows utilities to be compared across nests; without the rescaling, utilities can be compared only for goods within the same nest. For a further discussion of the

What is I_{nk}? I_{nk} is called the inclusive value or inclusive utility for alternative k (nest B_k) in the first level. It is calculated as the log of the denominator of the second level. The inclusive value links the two levels of the nested logit model by bringing information from the bottom level into the upper level. Essentially, $\lambda_k I_{nk}$ captures the expected value or utility to individual n of the alternatives available in nest B_k. The coefficient λ_k of I_{nk} is called the log-sum coefficient. Recall that λ_k reflects the degree of independence among the unobserved portions of utility for alternatives in nest B_k. It is important that the inclusive value enters as an independent variable in the first level. Note that the probability of choosing nest B_k in the first level depends on the expected utility that the decision maker receives from choosing that nest. Well, this expected utility is made up of the utility he receives no matter which alternative he chooses in the nest i.e. $W_{nk} = w_{nk}\gamma$ and the expected extra utility that he receives from being able to choose the best alternative in the nest, which is $\lambda_k I_{nk}$.

2.2 Estimation

To estimate nested logit models in STATA 10, you use the NLOGIT command. Before you can use this command, though, you first need to create a variable that defines the structure of the decision tree using the NLOGITGEN command. Let's return to our Dutch elections example. Suppose that Dutch voters first decide whether to vote for a religious party (CDA) or a secular party Pvda, VVD, D66). If they decide to vote for a religious party, then they vote for the CDA. If they decide to vote for a secular party, then they must choose between the Pvda, VVD, or D66. This decision tree is shown in Figure 3.

Figure 3: Nested Logit Model for Dutch Elections



To do this, we would type:

```
nlogitgen type= vote(religious: 2, secular: 1|3|4);
```

This says create a variable TYPE that defines the structure of the decision tree. The two branches in the initial level are called RELIGIOUS and SECULAR. Party 2 is available if you go down the religious branch, and parties 1, 3, and 4 are available if you go down the secular branch.

differences between these two models, see Train (2007, 88) and Heiss (2002). Note that STATA 9 only allows you to estimate the non-normalized nested logit. In STATA 10, the default is to estimate the normalized nested logit model using the NLOGIT command; you can estimate the non-normalized model by using NONNORMALIZED as an option.

To see the structure of your tree, you can type:

```
nlogittree vote type, choice(choice);
```

where CHOICE is the dependent variable. The output following this command is:

```
. nlogittree vote type, choice(choice)

tree structure specified for the nested logit model

 type       N       vote  N    k
-----------------------------------
 religious 1239 --- 2    1239  454
 secular   3717 --- 1    1239  449
                 |- 3    1239  192
                 +- 4    1239  144
-----------------------------------
                  total  4956 1239

k = number of times alternative is chosen
N = number of observations at each level
```

Now that we have defined two different types of parties, we can estimate our nested logit model. Suppose that we believe that RELIGION (an individual-specific variable) influences whether you will vote for a religious or secular party (top level) and that various policy dimensions (alternative-specific variable) influence the choice of party at the bottom level. We would type the following:

```
nlogit choice right abortion incomedifference nuclear
    || type: religious, base(secular)
    || vote:,  noconstant case(respid);
```

The delimiters || separate equations. The first equation specifies the dependent variable CHOICE and four alternative-specific variables RIGHT, ABORTION, INCOMEDIFFERENCE, NUCLEAR that capture the distance from the voter to each party on a number of policy dimensions. These are alternative-specific variables because they vary among the bottom-level alternatives. We obtain one parameter for each of these variables. These estimates are listed in the equation subtable labeled VOTE. For the second equation, we specify the variable TYPE; this variable identifies the first-level alternatives i.e. whether a party is religious or secular. Following the colon after TYPE, I have specified one individual-specific variable RELIGION which captures how religious the voter is. We would obtain a parameter estimate for each variable for each alternative at this level except that one set of parameters for one alternative is set to 0. In this case, I set the baseline category to be SECULAR parties. The variable identifying the bottom-level alternatives, VOTE, is specified after the second equation delimiter. The NOCONSTANT option suppresses bottom-level alternative-specific constant terms. The CASE() tells us the variable that defines individual voters.

Below, I show part of the output that is produced in STATA. The coefficients can be interpreted in the usual manner. The positive coefficient on RELIGION indicates that religious individuals are more likely to vote for religious parties than secular parties. The negative coefficients on RIGHT, ABORTION, INCOMEDIFFERENCE, and NUCLEAR indicate that individuals are less likely to vote for parties that are further from them on these issue dimensions. At the bottom of the table, you will find the coefficient (λ_k) on the inclusive value (I_{nk}). You will remember that this coefficient is sometimes called the log-sum coefficient or the dissimilarity parameter. Recall that λ_k reflects the degree of independence among the unobserved portions of utility for the alternatives in each nest. A high λ_k means greater independence and less correlation i.e. the alternatives in the nest are dissimilar (for unobserved reasons). For our model to be consistent with a random utility model, it must be the case that $0 < \lambda_k < 1 \; \forall k$; it is.

```
RUM-consistent nested logit regression         Number of obs      =       2908
Case variable: respid                          Number of cases    =        727

Alternative variable: vote                     Alts per case: min =          4
                                                              avg =        4.0
                                                              max =          4

                                                  Wald chi2(5)    =     172.72
Log likelihood = -552.56599                       Prob > chi2     =     0.0000

------------------------------------------------------------------------------
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
vote         |
       right |  -.5806895   .0567767   -10.23   0.000    -.6919699   -.4694091
    abortion |  -.2594062   .0380167    -6.82   0.000    -.3339176   -.1848949
incomediff~e |  -.3631627   .0486444    -7.47   0.000     -.458504   -.2678214
     nuclear |  -.0968508   .0405587    -2.39   0.017    -.1763444   -.0173572
-------------+----------------------------------------------------------------
type equations
-------------+----------------------------------------------------------------
religious    |
   religious |   .9916663   .1485749     6.67   0.000     .7004648    1.282868
-------------+----------------------------------------------------------------
secular      |
   religious |     (base)
------------------------------------------------------------------------------
dissimilarity parameters
-------------+----------------------------------------------------------------
type         |
/religious~u |          1     366348                      -718027.8    718029.8
/secular_tau |   .8288778    .088623                         .65518    1.002576
------------------------------------------------------------------------------
LR test for IIA (tau = 1):           chi2(2) =     3.13   Prob > chi2 = 0.2092
------------------------------------------------------------------------------
```

Note that $\lambda_k = 1$ (religious_tau=1) for the first nest since there is only one party in the religious party nest. To take the degenerate nature of this into account, you should actually constrain $\lambda_{religious} = 1$ by using STATA's CONSTRAINT command:[4]

[4]You can also use the CONSTRAINT command to constrain the λ_k's to be the same across all or some of the nests if you wanted.

```
constraint 1 [religious_tau]_cons=1;
nlogit choice right abortion incomedifference nuclear
    || type: religious, base(secular)
    || vote:,  noconstant case(respid) constraints(1);
```

The output from this is shown below:

```
RUM-consistent nested logit regression         Number of obs      =       2908
Case variable: respid                          Number of cases    =        727

Alternative variable: vote                     Alts per case: min =          4
                                                              avg =        4.0
                                                              max =          4

                                                  Wald chi2(5)    =     172.72
Log likelihood = -552.56599                       Prob > chi2     =     0.0000

 ( 1)  [religious_tau]_cons = 1
------------------------------------------------------------------------------
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
vote         |
       right |  -.5806895   .0567767   -10.23   0.000    -.6919699   -.4694091
    abortion |  -.2594062   .0380167    -6.82   0.000    -.3339176   -.1848949
incomediff~e |  -.3631627   .0486444    -7.47   0.000     -.458504   -.2678214
     nuclear |  -.0968508   .0405587    -2.39   0.017    -.1763444   -.0173572
------------------------------------------------------------------------------
type equations
------------------------------------------------------------------------------
religious    |
   religious |   .9916663   .1485749     6.67   0.000     .7004648    1.282868
-------------+----------------------------------------------------------------
secular      |
   religious |     (base)
------------------------------------------------------------------------------
dissimilarity parameters
------------------------------------------------------------------------------
type         |
/religious~u |          1          .                           .           .
/secular_tau |   .8288778    .088623                      .65518    1.002576
------------------------------------------------------------------------------
LR test for IIA (tau = 1):           chi2(1) =     3.13   Prob > chi2 = 0.0769
------------------------------------------------------------------------------
```

Remember that if $\lambda_k = 1 \; \forall k$, then we have complete independence and the nested logit reduces to a standard logit model. An LR test to see if this is the case is reported at the bottom of the table with the null being that $\lambda_k = 1 \; \forall k$. Note that when we incorrectly failed to constrain religious_tau=1, the LR test indicates that we cannot reject the null that all the all of the log-sum coefficients are 1 ($p = 0.2092$) and hence we can use a standard logit rather than a nested logit. However, once we appropriately constrain religious_tau=1, we find that we can reject the null that all of the log-sum coefficients are 1, at least at the 90% level ($p = 0.0769$). This indicates that a nested logit is, in fact, probably more appropriate here than a standard logit model.

Note that the joint test that $\lambda_k = 1 \; \forall k$ can also be seen as a test of the IIA assumption; if $\lambda_k = 1 \; \forall k$, then IIA holds and it is appropriate to use a standard logit model. One word of warning should be given about this particular test of IIA – the results will depend on exactly how you are specified the decision tree. In other words, different specifications of the decision tree can lead to conflicting results. The Hausman-McFadden test of IIA from last week is superior because it does not depend on the tree structure specified in the nested logit model.

2.3 Interpretation

One useful thing to do when interpreting the results is to look at a summary of the alternatives and frequencies in the estimation sample. After estimating the nested logit model, you can type ESTAT ALTERNATIVES and you will get the following:

```
estat alternatives

 Alternatives summary for type
+-------------------------------------------------------------------------+
|          Alternative                    |    Cases   Frequency    Percent |
|  index        value             label   |   present   selected   selected |
|-----------------------------------------+---------------------------------|
|      1            1          religious  |       727        243      33.43 |
|      2            2            secular  |      2181        484      66.57 |
+-------------------------------------------------------------------------+

 Alternatives summary for vote
+-------------------------------------------------------------------------+
|          Alternative                    |    Cases   Frequency    Percent |
|  index        value             label   |   present   selected   selected |
|-----------------------------------------+---------------------------------|
|      1            1                 1   |       727        271      37.28 |
|      2            2                 2   |       727        243      33.43 |
|      3            3                 3   |       727        115      15.82 |
|      4            4                 4   |       727         98      13.48 |
+-------------------------------------------------------------------------+
```

You can obviously use the equations in the notes to calculate the same quantities of interest that we always do. For example, you can calculate (i) the probability (P1) of choosing one of the alternatives in the top level, Pr(Type), (ii) the probability (P2) of choosing one of the alternatives in the bottom level, Pr(Vote), (iii) the probability (CONDP) of choosing an alternative in the bottom level given a particular choice at the top level, Pr(Vote|Type), or (iv) the value of the inclusive value (IV) for different scenarios. STATA has a variety of post-estimation commands that will calculate these quantities for the values associated with each of the observations in the sample. For example, you could type:

```
predict p*;
predict condp, condp hlevel(2);
predict iv, iv;
```

Below is output illustrating what has been calculated for the first four observations in the data set.

```
. sort respid  vote type

. list respid vote type choice p1 p2 condp iv in 1/16, sepby(respid) divider
```

	respid	vote	type	choice	p1	p2	condp	iv
1.	1	1	secular	1	.9944792	.5472152	.550253	-1.284101
2.	1	2	religious	0	.0055208	.0055208	1	-6.25806
3.	1	3	secular	0	.9944792	.035428	.0356246	-1.284101
4.	1	4	secular	0	.9944792	.411836	.4141223	-1.284101
5.	2	1	secular	0	.2920647	.24244	.83009	-1.061004
6.	2	2	religious	1	.7079353	.7079353	1	.0059347
7.	2	3	secular	0	.2920647	.0305016	.1044344	-1.061004
8.	2	4	secular	0	.2920647	.0191231	.0654756	-1.061004
9.	3	1	secular	0	.9377829	.0962957	.1026844	-.1181727
10.	3	2	religious	0	.0622171	.0622171	1	-2.81084
11.	3	3	secular	0	.9377829	.0060127	.0064116	-.1181727
12.	3	4	secular	1	.9377829	.8354746	.8909039	-.1181727
13.	4	1	secular	0	.2045946	.0053378	.0260897	-.5585938
14.	4	2	religious	0	.7954054	.7954054	1	.8948154
15.	4	3	secular	1	.2045946	.0837598	.4093938	-.5585938
16.	4	4	secular	0	.2045946	.115497	.5645165	-.5585938

Let's look at individual 1. Given the values on the independent variables for individual 1, the probability that he chooses a secular party is 0.994 and the probability that he chooses a religious party is 0.005. That this individual is much more likely to choose a secular party rather than a religious party can also be seen from the inclusive value – it is -1.28 for the secular party nest but -6.26 for the religious party nest. Recall that the inclusive value multiplied by the log-sum coefficient ($\lambda_k I_{nk}$) captures the expected utility to the individual of the alternatives available in nest k. Clearly, individual one has a higher expected utility from the secular party nest than the religious party nest. Note that this is reflected in his choice to vote for one of the secular parties. The probability that individual 1 votes each of the four parties is shown by P2. In other words, individual 1 votes for party 1 with a probability of 0.55, for party 2 with a probability of 0.006, and so on. The probability that individual 1 votes for a particular party given his choice to pick from among secular or religious parties is shown by CONDP. The probability that individual 1 votes for party 1 if he has chosen to vote for a secular party is 0.55. Similarly, the probability that individual 1 votes for party 2 if he has chosen to vote for a religious party is 1.

2.4 Extensions

2.4.1 Three-Level Nested Logit

You can estimate nested logit models with more than two levels. For example, a three-level nested logit can be obtained by partitioning the choice set into nests and then the nests into sub-nests. As before, you can

express the choice probabilities as a series of logits. The top level describes the choice of nest; the second level describes the choice of subnest; and the bottom level describes the choice of alternative within each subnest. As you would expect, the top level includes an inclusive value for each nest and this captures the expected utility that the decision maker gets from the subnests within the nest; it is calculated as the log of the denominator of the second level model. Similarly, the second level models include an inclusive value for each subnest, which represents the expected utility of the alternatives in each subnest; it is calculated as the log of the denominator of the third level model.

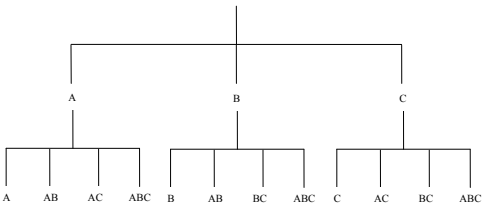
Each layer of nesting introduces parameters that capture the degree of correlation among alternatives within the nests. With the full choice set partitioned into nests, the parameter λ_k is introduced for nest k. If the nests are partitioned into additional subnests, then a parameter σ_{mk} is introduced for subnest m in nest k. Using the decomposition of the probability into a series of logits, σ_{mk} is the coefficient of the inclusive value in the second level model and $\lambda_k \sigma_{mk}$ is the coefficient of the inclusive value in the first level model.

Nested logit models with more than two-levels can be estimated in STATA 10 using the same commands as outlined earlier. For more details on this, see the NLOGIT command in the STATA manuals.

2.4.2 Generalized Nested Logit

Each of the alternatives in the nested logit models that we have examined so far only appear in one nest. However, this is not always the case - alternatives may appear in more than one nest. For example, you might have the following type of structure:

Figure 4: Overlapping Nests in a Nested Logit Model with 2 Levels



It is possible to construct a generalized nested logit model or cross-nested logit model to deal with this type of situation (Wen & Koppelman 2001, Hess, Bierlaire & Polak 2004, Vovsha 1997).[5] Let the nests of alternatives be labeled $B_1, B_2 \ldots, B_K$ as before. Each alternative can be a member of more than one nest. In fact, an alternative can be in a nest to differing degrees. As Train (2007, 95) notes, an alternative is allocated among nests, with the alternatives being in some nests more than others. An allocation parameter α_{jk} reflects the extent to which alternative j is in nest k. This parameter must be non-negative: $\alpha_{jk}\ \forall j, k$. The allocation parameters should be constrained so that they sum to one: $\sum_k \alpha_{jk} = 1\ \forall j$. In this way, α_{jk} reflects the portion of the alternative that is allocated to each nest. A parameter λ_k is defined for each nest as in the standard nested logit model.

[5] Unfortunately, there are no canned routines for these models at the moment in STATA.

The probability that individual n chooses alternative i from the choice set is:

$$P_{ni} = \frac{\sum_k \left(\alpha_{ik} e^{V_{ni}}\right)^{1/\lambda_k} \left(\sum_{j \in B_k} \left(\alpha_{jk} e^{V_{nj}}\right)^{1/\lambda_k}\right)^{\lambda_k - 1}}{\sum_{\iota=1}^{K} \left(\sum_{j \in B_\iota} \left(\alpha_{j\iota} e^{V_{nj}}\right)^{1/\lambda_\iota}\right)^{\lambda_\iota}} \tag{12}$$

As before, this can be decomposed into two logits - the probability of choosing nest B_k multiplied by the probability of choosing alternative i in nest k:

$$P_{nB_k i} = \frac{\left(\sum_{j \in B_k} \left(\alpha_{jk} e^{V_{nj}}\right)^{1/\lambda_k}\right)^{\lambda_k}}{\sum_{\iota=1}^{K} \left(\sum_{j \in B_\iota} \left(\alpha_{j\iota} e^{V_{nj}}\right)^{1/\lambda_\iota}\right)^{\lambda_\iota}} \times \frac{\left(\alpha_{ik} e^{V_{ni}}\right)^{1/\lambda_k}}{\sum_{j \in B_k} \left(\alpha_{jk} e^{V_{nj}}\right)^{1/\lambda_k}} \tag{13}$$

3 Probit Models

Recall from earlier that there were at least three limitations of of standard logit models: (i) they cannot represent random taste variation, (ii) they only allow restrictive substitution patterns (IIA), and (iii) they cannot be used with panel data when unobserved factors are correlated over time.[6] Probit models deal with all three of these problems. The only limitation to probit models is that they require normal distributions for all of the unobserved portions of utility.

The probit model is derived under the assumption that the unobserved components of utility are distributed jointly normal:

$$\epsilon_{nj} \sim \text{MVN}(0, \Omega) \tag{14}$$

As always, we start with our basic utility equation:

$$U_{nj} = V_{nj} + \epsilon_{nj} \; \forall j \tag{15}$$

Consider the vector composed of each ϵ_{nj}, labeled $\epsilon_n' = \langle \epsilon_{n1}, \ldots, \epsilon_{nJ} \rangle$. As mentioned above, we assume that ϵ_n is distributed normal with a mean vector of zero and a covariance matrix Ω. The density of ϵ_n is:

$$f(\epsilon_n) = \phi(\epsilon_n) = \frac{1}{(2\pi)^{J/2} |\Omega|^{1/2}} e^{-1/2 \epsilon_n' \Omega^{-1} \epsilon_n} \tag{16}$$

The choice probability for a probit model is:

$$\begin{aligned} P_{ni} &= \text{Prob}(V_{ni} + \epsilon_{ni} > V_{nj} + \epsilon_{nj} \;\; \forall j \neq i) \\ &= \int_\epsilon I(V_{ni} + \epsilon_{ni} > V_{nj} + \epsilon_{nj} \;\; \forall j \neq i) \phi(\epsilon_n) d\epsilon_n \end{aligned} \tag{17}$$

where $I(\cdot)$ is an indicator of whether the statement in parentheses holds and the integral is over all values of ϵ_n. Unlike with standard logit models, this integral does not have a closed form solution and must be

[6] As before, I will not focus on probit models with panel data in what follows.

evaluated numerically through simulation; we'll look at how this is done in a moment.

Note that Eq. (17) is a J-dimensional integral over the J errors. However, as we have seen before, only differences in utility matter and so choice probabilities can be expressed in terms of error differences. If we difference against alternative i, then we have $\tilde{U}_{nji} = U_{nj} - U_{ni}$, $\tilde{V}_{nji} = V_{nj} - V_{ni}$, and $\tilde{\epsilon}_{nji} = \epsilon_{nj} - \epsilon_{ni}$. We now have $P_{ni} = \text{Prob}(\tilde{U}_{nji} < 0 \ \ \forall j \neq i \ )$. Define the vector $\tilde{\epsilon}_{ni} = \langle \tilde{\epsilon}_{n1i}, \ldots, \epsilon_{nJi} \rangle$ where "..." is over all alternatives except i, so that $\tilde{\epsilon}_{ni}$ now has J-1 dimensions. Since the difference between two normals is normal, the density of the error differences is:

$$\phi(\tilde{\epsilon}_{ni}) = \frac{1}{(2\pi)^{1/2(J-1)}|\tilde{\Omega}_i|^{1/2}} e^{-1/2\tilde{\epsilon}'_{ni}\tilde{\Omega}^{-1}\tilde{\epsilon}_{ni}} \tag{18}$$

where $\tilde{\Omega}_i$ is the covariance of $\tilde{\epsilon}_{ni}$ and is derived from Ω.[7] Now the choice probability is:

$$P_{ni} = \int_{\epsilon} I(\tilde{V}_{nji} + \tilde{\epsilon}_{nji} < 0 \ \ \forall j \neq i)\phi(\tilde{\epsilon}_{ni})d\tilde{\epsilon}_{ni} \tag{22}$$

which is a (J-1)-dimensional integral over all the values of the error differences.

3.1 Identification

As we have seen, all discrete choice models need to be normalized to take into account the fact that the level and scale of utility is irrelevant. In logit and nested logit models, the necessary normalization for scale and level occurs automatically with the distributional assumptions that are placed on the error terms. However, things are not so obvious with probit models and care needs to be taken to make sure that the model is

[7] How do we get $\tilde{\Omega}_i$? Train (2007, 103-104) indicates how $\tilde{\Omega}_i$ can be obtained from Ω with the help of matrix M_i. Let M_i be a J-1 identity matrix with an extra column of -1's added as the i^{th} column. If we had four alternatives and $i = 3$, we have:

$$M_i = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix} \tag{19}$$

We can use the M_i matrix to transform the covariance matrix of error terms into the covariance matrix of error differences: $\tilde{\Omega}_i = M_i \Omega M_i'$. To see how this works, imagine a situation with three alternatives and errors $\langle \epsilon_{n1}, \epsilon_{n2}, \epsilon_{n3} \rangle$ with a covariance matrix:

$$\Omega = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \tag{20}$$

If we take differences against alternative 2, we have:

$$\begin{aligned} \tilde{\Omega}_2 &= \text{cov} \begin{bmatrix} \epsilon_{n1} - \epsilon_{n2} \\ \epsilon_{n3} - \epsilon_{n2} \end{bmatrix} \\ &= \begin{bmatrix} 1 & -1 & 0 \\ 0 & -1 & 1 \end{bmatrix} \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} \\ \sigma_{21} & \sigma_{22} & \sigma_{23} \\ \sigma_{31} & \sigma_{32} & \sigma_{33} \end{bmatrix} \begin{bmatrix} 1 & 0 \\ -1 & -1 \\ 0 & 1 \end{bmatrix} \\ &= \begin{bmatrix} \sigma_{11} + \sigma_{22} - 2\sigma_{12} & \sigma_{13} + \sigma_{22} - \sigma_{12} - \sigma_{23} \\ \sigma_{13} + \sigma_{22} - \sigma_{12} - \sigma_{23} & \sigma_{33} + \sigma_{22} - 2\sigma_{23} \end{bmatrix} \end{aligned} \tag{21}$$

identified. The key issue with identification is that parameters that do not affect the choice of decision makers cannot be estimated. In an unnormalized model, there can be parameters related to the scale and level of utility that are unidentified and cannot be estimated. By normalizing the model, we get rid of these parameters. But which parameters are related to the scale and level of utility? Train (2007, 105) provides a procedure that can always be used to normalize a probit model and ensure identification. This procedure can also be used to check whether other normalization procedures lead to identified models.

Start with a choice scenario with four alternatives: $U_{nJ} = V_{nj} + \epsilon_{nj}$, $j = 1, \ldots, 4$. The vector of errors is $\epsilon_n' = \langle \epsilon_{n1}, \ldots, \epsilon_{n4} \rangle$. It is normally distributed with mean 0 and the following covariance matrix:

$$\Omega = \begin{bmatrix} \sigma_{11} & \sigma_{12} & \sigma_{13} & \sigma_{14} \\ \cdot & \sigma_{22} & \sigma_{23} & \sigma_{24} \\ \cdot & \cdot & \sigma_{33} & \sigma_{34} \\ \cdot & \cdot & \cdot & \sigma_{44} \end{bmatrix} \tag{23}$$

There are 10 distinct elements in this matrix. With J alternatives, you will have $J(J+1)/2$ distinct elements in the covariance matrix of errors. To deal with the fact that the level of utility is irrelevant, we can take utility differences. Let's take differences with respect to the first alternative. Let the error differences be $\tilde{\epsilon_{nj1}} = \epsilon_{nj} - \epsilon_{n1}$ for $j = 1, \ldots, 4$. Thus, we have a vector of error differences $\tilde{\epsilon}_{n1} = \langle \tilde{\epsilon}_{n21}, \tilde{\epsilon}_{n31}, \tilde{\epsilon}_{n41} \rangle$. The covariance matrix for the error differences is:

$$\tilde{\Omega}_1 = \begin{bmatrix} \theta_{22} & \theta_{23} & \theta_{24} \\ \cdot & \theta_{33} & \theta_{34} \\ \cdot & \cdot & \theta_{44} \end{bmatrix} \tag{24}$$

where

$$\begin{aligned} \theta_{22} &= \sigma_{22} + \sigma_{11} - 2\sigma_{12} \\ \theta_{33} &= \sigma_{33} + \sigma_{11} - 2\sigma_{13} \\ \theta_{44} &= \sigma_{44} + \sigma_{11} - 2\sigma_{14} \\ \theta_{23} &= \sigma_{23} + \sigma_{11} - \sigma_{12} - \sigma_{13} \\ \theta_{24} &= \sigma_{24} + \sigma_{11} - \sigma_{12} - \sigma_{14} \\ \theta_{34} &= \sigma_{34} + \sigma_{11} - \sigma_{13} - \sigma_{14} \end{aligned} \tag{25}$$

We have already seen how to obtain $\tilde{\Omega}_1$ using the M_i matrix. To set the scale of utility, we can normalize one of the diagonal elements to 1 so that we have:

$$\tilde{\Omega}_1^* = \begin{bmatrix} 1 & \theta_{23}^* & \theta_{24}^* \\ \cdot & \theta_{33}^* & \theta_{34}^* \\ \cdot & \cdot & \theta_{44}^* \end{bmatrix} \tag{26}$$

where

$$\theta^*_{33} = \frac{\sigma_{33} + \sigma_{11} - 2\sigma_{13}}{\sigma_{22} + \sigma_{11} - 2\sigma_{12}}$$
$$\theta^*_{44} = \frac{\sigma_{44} + \sigma_{11} - 2\sigma_{14}}{\sigma_{22} + \sigma_{11} - 2\sigma_{12}}$$
$$\theta^*_{23} = \frac{\sigma_{23} + \sigma_{11} - \sigma_{12} - \sigma_{13}}{\sigma_{22} + \sigma_{11} - 2\sigma_{12}}$$
$$\theta^*_{24} = \frac{\sigma_{24} + \sigma_{11} - \sigma_{12} - \sigma_{14}}{\sigma_{22} + \sigma_{11} - 2\sigma_{12}}$$
$$\theta^*_{34} = \frac{\sigma_{34} + \sigma_{11} - \sigma_{13} - \sigma_{14}}{\sigma_{22} + \sigma_{11} - 2\sigma_{12}} \tag{27}$$

There are now five elements in $\tilde{\Omega}_1$. These five elements are the only identified parameters in the model. This number is obviously less than the 10 elements in the original error covariance matrix, Ω. As a result, it is not possible to solve for all 10 of the σ's from the 5 θ^*'s. As we noted at the beginning, with J alternatives, an unrestricted covariance matrix will have $[(J-1)J/2]-1$ covariance parameters when normalized compared to the $J(J+1)/2$ parameters when unnormalized. This reduction is not a restriction since we have simply got rid of irrelevant parameters that are related to the scale and level of utility. In other words, only the five elements in $\tilde{\Omega}^*_1$ provide information about the variance and covariance of the errors that is independent of scale and level. The model is now exactly identified. To estimate this model in STATA, you would use the ASMPROBIT command.[8]

The procedure that has just been outlined will always lead to an identified model. However, some scholars prefer to place their own restrictions on the covariance matrix of the error terms, Ω, perhaps based on some theoretical intuition.[9] The question, though, is whether these restrictions lead to an identified model. To see if they do lead to an identified model, you would first create the covariance matrix of error differences, $\tilde{\Omega}_1$. Then, you would normalize for scale and create the normalized covariance matrix of error differences, $\tilde{\Omega}^*_1$, as we just did above. Then you need to see whether you can calculate the original restricted elements of the covariance matrix of error terms, Ω, from the estimated elements of the normalized covariance matrix of error differences, $\tilde{\Omega}^*_1$. If you can, the restrictions have led to an identified model. If you cannot, then the restrictions are not sufficient to produce an identified model. Train (2007, 108-110) gives examples of restrictions that lead to an identified model and restrictions that do not. As an analyst, you need to check that you have an identified model if you place restrictions on the covariance matrix of error terms.

3.2 Taste Variation: Random-Coefficients Model

Probit is able to take account of random taste variation so long as the random coefficients are normally distributed. Suppose that the utility function is linear in parameters as before but that the coefficients now vary randomly over decision makers. In effect, we have:

$$U_{nj} = x_{nj}\beta_n + \epsilon_{nj} \tag{28}$$

[8] It is worth noting that the default for the ASMPROBIT command in STATA is to set the scale of utility by normalizing one of the diagonal elements to 2 rather than 1 as I did in the example just given i.e. STATA sets $\theta_{22} = 2$.

[9] You can do this in STATA using the ASMPROBIT command and the STRUCTURAL() option.

where β_n is a vector of coefficients for individual n indicating his tastes for the alternatives. Let β_n be normally distributed with mean b and covariance W : $\beta_n \sim N(b, W)$. We now want to estimate b and W. Note that the utility function can be rewritten with β_n decomposed into its mean and deviations from its mean:

$$U_{nj} = x_{nj}b + x_{nj}\tilde{\beta}_n + \epsilon_{nj} \tag{29}$$

where $\tilde{\beta}_n = \beta_n - b$. The last two terms in the utility equation are random; denote their sum to be e_{nj} so that we have:

$$e_{nj} = x_{nj}\tilde{\beta}_n + \epsilon_{nj} \tag{30}$$

From this, we have the following utility function:

$$\begin{aligned} U_{nj} &= x_{nj}b + (x_{nj}\tilde{\beta}_n + \epsilon_{nj}) \\ &= x_{nj}b + e_{nj} \end{aligned} \tag{31}$$

The covariance of e_{nj} depends on both W and the x_{nj}'s and so the covariance differs over decision makers.

Consider this setup for the case where we have two alternatives. We have:

$$\begin{aligned} U_{n1} &= x_{n1}\beta_n + \epsilon_{n1} \\ U_{n2} &= x_{n2}\beta_n + \epsilon_{n2} \end{aligned}$$

Let β_n be distributed normally with mean b and variance σ_β and let ϵ_{nj} be distributed iid with mean 0 and variance σ_ϵ. We would now rewrite the utility functions as:

$$\begin{aligned} U_{n1} &= x_{n1}b + e_{n1} \\ U_{n2} &= x_{n2}b + e_{n2} \end{aligned}$$

where e_{n1} and e_{n2} are jointly normally distributed. Each of these new error terms has a zero mean:

$$E(e_{nj}) = E(x_{nj}\tilde{\beta}_n + \epsilon_{nj}) = 0 \tag{32}$$

The variance for each of the new error terms is:

$$V(e_{nj}) = V(x_{nj}\tilde{\beta}_n + \epsilon_{nj}) = x_{nj}^2\sigma_\beta + \sigma_\epsilon \tag{33}$$

The covariance of the new error terms is:

$$\begin{aligned} \text{cov}(e_{n1}, e_{n2}) &= E[(x_{n1}\tilde{\beta}_n + \epsilon_{n1})(x_{n2}\tilde{\beta}_n + \epsilon_{n2})] \\ &= E(x_{n1}x_{n2}\tilde{\beta}_n^2 + \epsilon_{n1}\epsilon_{n2} + \epsilon_{n1}x_{n2}\tilde{\beta}_n + \epsilon_{n2}x_{n1}\tilde{\beta}_n) \\ &= x_{n1}x_{n2}\sigma_\beta \end{aligned} \tag{34}$$

The covariance matrix is:

$$\Omega = \begin{bmatrix} x_{n1}^2\sigma_\beta + \sigma_\epsilon & x_{n1}x_{n2}\sigma_\beta \\ x_{n1}x_{n2}\sigma_\beta & x_{n2}^2\sigma_\beta + \sigma_\epsilon \end{bmatrix}$$

$$= \sigma_\beta \begin{bmatrix} x_{n1}^2 & x_{n1}x_{n2} \\ x_{n1}x_{n2} & x_{n2}^2 \end{bmatrix} + \sigma_\epsilon \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{35}$$

We now need to set the scale of utility. The typical normalization is to set $\sigma_\epsilon = 1$ so that we have:

$$\Omega = \sigma_\beta \begin{bmatrix} x_{n1}^2 & x_{n1}x_{n2} \\ x_{n1}x_{n2} & x_{n2}^2 \end{bmatrix} + \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{36}$$

We then estimate parameters b and σ_b. In effect, the analyst gets to estimate the mean (b) and the variance (σ_b) of the random coefficient. This is called a random coefficients model. Random coefficients models are obviously good for examining random taste variation since they allow for the study of heterogeneity in the impact of the independent variables on the dependent variable.

To my knowledge, no one has used a random-coefficients probit model in political science, though Rivers (1988) did employ a similar approach in one article. Hausman and Wise (1978) were the first to actually derive this model. To see this model in use, consult Haaijer et al. (1998).

The random-coefficients probit model cannot be estimated in STATA 10.

3.3 Substitution Patterns and the Independence of Irrelevant Alternatives

Unlike standard logit models, probit models do not exhibit the restrictive IIA property. This has traditionally been the main reason why analysts switch from logit models to probit models. Unlike logit models, probit models can exhibit any substitution pattern among alternatives. Obviously, different covariance matrices Ω produce different substitution patterns. But by estimating the covariance matrix in a probit model, the analyst can essentially determine the substitution pattern that is most appropriate for the data that is being examined. The analyst basically has two choices: (i) estimate a full covariance matrix or (ii) impose particular restrictions on the covariance matrix to represent specific substitution patterns.

3.3.1 Full Covariance Matrix

Essentially, when an analyst estimates a full covariance matrix, he is choosing to implement the identification procedure that I outlined above. You start with a full covariance matrix, Ω, of the form shown in Eq. (53). You then normalize for the scale and level of utility to get the normalized covariance matrix, $\tilde{\Omega}_1^*$, shown in

Eq. (37).

$$\tilde{\Omega}_1^* = \begin{bmatrix} 1 & \theta_{23}^* & \theta_{24}^* \\ \cdot & \theta_{33}^* & \theta_{34}^* \\ \cdot & \cdot & \theta_{44}^* \end{bmatrix} \tag{37}$$

You then estimate the elements of $\tilde{\Omega}_1^*$. The estimated values of $\tilde{\Omega}_1^*$ can capture any substitution pattern; the normalization process does not eliminate any substitution patterns because we are only getting rid of things that are unrelated to the behavior of the decision maker.

The limitation of this approach is that the estimated values of the θ^*'s are not directly interpretable. As Train (2007, 112-113) points out, you might think to assume that the variance of the unobserved portion of utility for alternative 3 is necessarily greater than that for alternative 4 ($\sigma_{33} > \sigma_{44}$) if we find that $\theta_{33}^* > \theta_{44}^*$. However, this is not necessarily the case. It is possible for $\theta_{33}^* > \theta_{44}^*$ but for ($\sigma_{44} > \sigma_{33}$) if the covariance σ_{14} is sufficiently greater than σ_{13} (see Eq. (27)). Equally, the fact that, say, $\theta_{23}^* < 0$ does not necessarily mean that the unobserved component of utility associated with alternative 2 is negatively correlated with that of alternative 3 i.e. it is not necessarily the case that $\sigma_{23} < 0$ just because $\theta_{23}^* < 0$; it is possible for $\sigma_{23} > 0$ but for σ_{12} and σ_{12} to be sufficiently large that $\theta_{23}^* < 0$ (see Eq. (27)).

If you are particularly interested in interpreting the substitution patterns among alternatives, then, you might not want to estimate the full covariance matrix. By estimating a restricted covariance matrix, it is, as we will see, possible to obtain more interpretable information about substitution patterns. Of course, you have to assume that you have used appropriate restrictions to get this extra information.

3.3.2 Restricted Covariance Matrix

As mentioned above, it is possible to place restrictions on the original covariance matrix, Ω. These restrictions limit the ability of the model to represent various substitution patterns. But if the structure is correct, then the true substitution pattern can be represented by the restricted covariance matrix. To think about how a substitution structure can be imposed on a covariance matrix and why this might be useful, consider the following example from Train (2007, 113-114). Suppose we have a four-alternative situation in which three alternatives share some characteristic, z, and one does not. Suppose the unobserved component of utility consists of two parts: the individual's concern about characteristic z, which is labeled r_n, and all other unobserved factors:

$$e_{nj} = r_n z_j + \epsilon_{nj} \tag{38}$$

where $z_j = 1$ for three alternatives and $z_j = 0$ for the other alternative. Let r_n, the concern that individuals have about characteristic z, be normally distributed over individuals with variance σ, and let $\epsilon_{nj}\ \forall j$ be iid normal with zero mean and variance ω.[10]

[10]Note that if $r_n = 0\ \forall n$, then we have an independent probit model with the IIA property i.e. the MPROBIT command. This is equivalent to setting the covariance matrix in Eq. (53) to be equal to the identity matrix i.e. the errors have unit variance and are uncorrelated.

The covariance matrix for $\epsilon_n = \langle \epsilon_{n1}, \ldots, \epsilon_{n4} \rangle$ is:

$$\Omega = \begin{bmatrix} 0 & 0 & 0 & 0 \\ \cdot & \sigma & \sigma & \sigma \\ \cdot & \cdot & \sigma & \sigma \\ \cdot & \cdot & \cdot & \sigma \end{bmatrix} + \omega \begin{bmatrix} 1 & 0 & 0 & 0 \\ \cdot & 1 & 0 & 0 \\ \cdot & \cdot & 1 & 0 \\ \cdot & \cdot & \cdot & 1 \end{bmatrix} \tag{39}$$

You would then calculate the covariance matrix for the error differences with respect to alternative 1. As Train indicates, this is:

$$\tilde{\Omega}_1 = \begin{bmatrix} \sigma & \sigma & \sigma \\ \cdot & \sigma & \sigma \\ \cdot & \cdot & \sigma \end{bmatrix} + \omega \begin{bmatrix} 2 & 1 & 1 \\ \cdot & 2 & 1 \\ \cdot & \cdot & 2 \end{bmatrix} \tag{40}$$

You would then normalize with respect to scale by setting $\sigma + 2\omega = 1$ so that we have:

$$\tilde{\Omega}_1^* = \begin{bmatrix} 1 & \theta & \theta \\ \cdot & 1 & \theta \\ \cdot & \cdot & 1 \end{bmatrix} \tag{41}$$

where $\theta = (\sigma + \omega)/(\sigma + 2\omega)$. As you can see, the choice of how to model the error term has led to particular restrictions on the normalized covariance matrix of the error differences. In effect, restrictions on the normalized covariance matrix of the error differences is equivalent to particular specifications for the error terms and vice versa; you can think of these restrictions in either way.

Note that the values of σ and ω cannot be calculated from θ. However, θ does tell us about the variance in utility due to z relative to that due to all other unobserved factors. If θ is estimated to be 0.75, then we get:

$$\begin{aligned} \theta &= 0.75 \\ \frac{\sigma + \omega}{\sigma + 2\omega} &= 0.75 \\ \sigma + \omega &= 0.75\sigma + 1.5\omega \\ 0.25\sigma &= 0.5\omega \\ \sigma &= 2\omega \end{aligned} \tag{42}$$

In other words, $\hat{\theta} = 0.75$ means that about two-thirds of the variance in the unobserved component of utility is due to z. As you can see, by restricting the covariance matrix, the estimated θ's become slightly more interpretable.

3.3.3 Random-Coefficient Models and Error-Components Models

Note that I basically used the same setup to illustrate random taste variation in probit models as I did to illustrate how analysts can estimate a restricted covariance matrix (compare Eq. (30) and Eq. (38)) . Essentially, I assumed that the unobserved component of utility consists of two parts: (i) a part that can vary over alternatives, individuals, or both, and (ii) a part that is IID over individuals and alternatives. In effect, I

had the following basic utility function:

$$\begin{aligned} U_{nj} &= x_{nj}\beta_j + (z_{nj}\eta_n + \epsilon_{nj}) \\ &= x_{nj}\beta_j + e_{nj} \end{aligned} \tag{43}$$

It is useful to get some terminology straight at this point. If all of the x's in Eq. (43) are the same as the z's in Eq. (43), then we have what is called a pure random-coefficients model. This is essentially what I described earlier when I looked at random taste variation in probit models. If none of the z's are x's in Eq. (43), then we have what is called a pure error-components model. This is how the probit model has traditionally been used in political science. The elements of z are assumed to be error components that introduce heteroskedasticity and correlation across alternatives in the unobserved portion of utility. Of course, it is possible to combine error-components and random-coefficients specifications i.e. elements of x that do not enter z are variables whose coefficients vary in the population with mean 0, and elements that enter both x and z are variables whose coefficients vary in the population with means represented by the appropriate elements in β. To learn more about the differences between these approaches, see Glasgow (2001).

3.4 Simulating Choice Probabilities

As we noted last week, probit choice probabilities do not have a closed form solution and must be approximated numerically. The most common method is to use simulation to approximate the probit choice probabilities.[11] Train (2007, 118-130, 240-261) outlines a number of simulators: accept-reject, smoothed accept-reject, and GHK. GHK is what STATA uses for its probit models.

3.4.1 Accept-Reject (AR) Simulator

The basic idea behind simulation methods can best be appreciated by looking at the accept-reject simulator. Recall that the probability that individual n chooses alternative i in a probit model is:

$$P_{ni} = \int_{\epsilon} I(V_{ni} + \epsilon_{ni} > V_{nj} + \epsilon_{nj} \; \forall j \neq i)\phi(\epsilon_n)d\epsilon_n \tag{44}$$

where $I(\cdot)$ is an indicator of whether the statement in parentheses is true or not, and $\phi(\epsilon_n)$ is the joint normal density with zero mean and covariance Ω. Essentially, what the simulator does is take draws of the random terms from their distributions. For each draw, determine whether those values of the errors, when combined with the observed variables faced by individual n, would result in alternative i being chosen. If so, the draw is called an accept; if the draw would result in some other alternative being chosen, then it is called a reject. The simulated probability is the proportion of draws that are accepts. Train outlines the steps of the AR simulator as follows:

[11] An alternative approach involves quadrature methods – they approximate the integral by a weighted function of specially chosen evaluation points. The problem with these methods is that they only really work when the dimension of the integral is quite small. For example, they only work well if the probit model has less than 4-5 alternatives.

1. Draw a value of the J-dimensional vector of errors, ϵ_n, from a normal density with zero mean and covariance, Ω. Label the draw ϵ_n^r with $r = 1$, and the elements of the draw as $\epsilon_{n1}^r, \ldots, \epsilon_{nJ}^r$.
2. Use the values of the errors to calculate the utility that each alternative is chosen with these errors. In other words, calculate $U_{nj}^r = V_{nj} + \epsilon_{nj}^r \; \forall j$.
3. Determine whether the utility of alternative i is greater than the utility for all of the other alternatives. In other words, calculate $I^r = 1$ if $U_{ni}^r > U_{nj}^r$, indicating an accept, and $I^r = 0$ otherwise, indicating a reject.
4. Repeat steps 1-3 many times. Label the number of repetitions R.
5. The simulated probability is the proportion of draws that are accepts: $\check{P}_{ni} = \frac{1}{R}\sum_{r=1}^{R} I^r$.

Thus, we have the integral $\int_\epsilon I(\cdot)\phi(\epsilon_n)d\epsilon_n$ approximated by $\frac{1}{R}\sum_{r=1}^{R} I^r$.

We would then take these simulated probabilities and use them in the context of maximum likelihood estimation. You will recall that the log-likelihood function is:

$$\ln\mathcal{L} = \sum_n \sum_j d_{nj} \ln P_{nj} \tag{45}$$

where $d_{nj} = 1$ if n chooses j and 0 otherwise. Since we are using simulated probabilities rather than the actual probabilities, we are using maximum simulated likelihood estimation (MSLE). In effect, we maximize the simulated log-likelihood function i.e.

$$\text{simulated } \ln\mathcal{L} = SLL = \sum_n \sum_j d_{nj} \ln \check{P}_{nj} \tag{46}$$

Although this simulation process is intuitive, there are some problems. One is that $\check{P}_{ni}$ can be zero for any finite number of draws i.e. it is possible for each of the R draws of the error terms to lead to a reject, and so the simulated probability would be 0. Having a zero probability is problematic because we are dealing with $\ln\check{P}_{nj}$, which is undefined. MSLE doesn't work if the simulated probability is zero for any decision maker. A second problem is that the simulated probabilities are not smooth in the parameters. The AR simulated probability is a step function. Recall that $\check{P}_{ni}$ is the proportion of draws for which alternative i has the highest utility. The problem is that a small change in a parameter is unlikely to change an accept to a reject or vice versa. As a result, $\check{P}_{ni}$ is constant for small changes in the parameter i.e. it is a step function. Within the range for which $\check{P}_{ni}$ does not change, the derivative with respect to the parameter is 0. This is problematic because we know that hill-climbing algorithms generally use the first derivative to know in which direction to climb and when they have reached the maximum.

3.4.2 Smoother AR Simulator

One way to solve the problems with the AR simulator is to replace the 0-1 AR indicator with a smooth, strictly positive function. Essentially, you start off as before by taking draws of the error terms and calculating the utility of each alternative for each draw: U_{nj}^r. However, instead of determining whether some

alternative i has the highest utility by using the indicator function I^r, you enter the simulated utilities into a function that is smooth and rises and falls with U^r_{nj}. One useful function is the logit function. This gives us the logit-smoothed AR simulator. Train outlines the steps as follows:

1. Draw a value of the J-dimensional vector of errors, ϵ_n, from a normal density with zero mean and covariance, Ω. Label the draw ϵ^r_n with $r = 1$, and the elements of the draw as $\epsilon^r_{n1}, \ldots, \epsilon^r_{nJ}$.
2. Use the values of the errors to calculate the utility that each alternative is chosen with these errors. In other words, calculate $U^r_{nj} = V_{nj} + \epsilon^r_{nj}\ \forall j$.
3. Put these utilities into the logit formula i.e.

$$S^r = \frac{e^{U^r_{ni}/\lambda}}{\sum_j e^{U^r_{nj}/\lambda}} \tag{47}$$

   where λ is a scale factor specified by the researcher that determines the degree of smoothing. As $\lambda \to 0$, S^r approaches the indicator function, I^r.
4. Repeat steps 1-3 many times. Label the number of repetitions R.
5. The simulated probability is the proportion of draws that are accepts: $\check{P}_{ni} = \frac{1}{R}\sum_{r=1}^{R} S^r$.

Thus, we have the integral $\int_\epsilon I(\cdot)\phi(\epsilon_n)d\epsilon_n$ approximated by $\frac{1}{R}\sum_{r=1}^{R} S^r$.

3.4.3 GHK Simulator

The GHK simulator is used by STATA and is the most widely used simulation process. The GHK simulator is more complicated than the simulators that I have just outlined, although the intuition is largely the same. To find out more about the GHK simulator, see Train (2007, 126-137).

3.4.4 Drawing from Densities

Recall that the objective of the simulation procedures outlined above is to approximate an integral of the form $\int t(\epsilon)f(\epsilon)d(\epsilon)$. One might think that we should use independent random draws from $f(\cdot)$ when using the simulation procedure. We can do this in STATA using the INTMETHOD(RANDOM) option.

However, it turns out that this is not always a good idea. In taking a sequence of draws from the density $f(\cdot)$, two issues come up: coverage and covariance. Coverage is straightforward and refers to the fact that the integral we are approximating is over the entire density f. Presumably, we want to evaluate $t(\epsilon)$ at values of ϵ that are spread throughout the domain of f. With independent random draws, it is possible for the draws to be clumped together. Procedures that produce better coverage should be preferred.

Covariance is the second issue. With independent random draws, the covariance over draws is zero. However, as Train (2007, 218) notes, the variance of the simulator is lower if the draws are negatively correlated

rather than independent. Moreover, if there is negative correlation between draws, then better coverage can be obtained. For example, with $R = 2$, it is possible for independent draws to both end up on the low side of the distribution. However, if negative correlation is induced, then the second draw will tend to be high if the first draw is low and vice versa; this provides better coverage.

For these two reasons – coverage and covariation – many scholars prefer not to use independent random draws for their simulation procedures but other methods that attain better coverage and induce negative correlation.

Antithetic Draws

One approach that improves coverage is to use antithetic draws (Hammersley & Morton 1956). The basic idea here is that we create various types of mirror images of a random draw. For example, with a symmetric distribution centered on zero, the simplest antithetic variate is created by reversing the sign of all elements of the draw. If a random draw is taken from $f(\epsilon)$ and the value $\epsilon_1 = \langle \epsilon_1^a, \epsilon_1^b \rangle$ is obtained, then the second draw should be $\epsilon_2 = \langle -\epsilon_2^a, -\epsilon_2^b \rangle$. To get R draws, you take $R/2$ independent draws from f and the other $R/2$ draws are created as the negative of the original draws. To use antithetic draws in STATA, use the ANTITHETICS option.

Halton Sequences

Halton sequences improve coverage and induce negative correlation. A Halton sequence is defined in terms of a (prime) number. Consider the following example where the prime number is 3. The Halton sequence for 3 is created by dividing the unit interval into three parts with breaks at 1/3 and 2/3. Then each of the three segments are divided into thirds, and the breakpoints for these segments are added to the sequences in a particular way. The sequence becomes $\frac{1}{3}$, $\frac{2}{3}$, $\frac{1}{9}$, $\frac{4}{9}$, $\frac{7}{9}$, $\frac{2}{9}$, $\frac{5}{9}$, $\frac{8}{9}$. Note that the lower breakpoints ($\frac{1}{9}$, $\frac{4}{9}$, $\frac{7}{9}$) in all three segments are entered in the sequence before the higher breakpoints ($\frac{2}{9}$, $\frac{5}{9}$, $\frac{8}{9}$). Then each of the 9 segments is divided into three parts etc. The process continues until the researcher has as many points as he wants. With a Halton sequence of 3, the sequence cycles over the unit interval every three numbers.

The Halton sequence is defined on the unit interval and can, therefore, be seen as well-placed draws from a standard uniform distribution. The Halton draws provide better coverage than random draws, on average, because they are created to increasingly fill in the unit interval evenly and ever more densely. Each cycle covers areas of the unit interval that have not previously been covered. The improved coverage and negative correlation make Halton draws more effective than random draws for simulation. Bhat (2001) shows that 100 Halton draws provide more precise results for a mixed logit model than 1000 random draws. To use Halton draws in STATA, use the INTMETHOD(HALTON) option.[12] You can specify the exact number of draws using the INTPOINTS(#) option; the default is 50.

[12]Rather than use Halton sequences, the default in STATA is to use Hammersley sequences; the idea is the same though.

3.4.5 Properties of Maximum Simulated Likelihood

Recall that the log-likelihood function is:

$$LL(\theta) = \sum_n \ln P_n(\theta) \tag{48}$$

where θ is a vector of parameters and P_n is the exact probability of the observed choice for observation n. The maximum likelihood (ML) estimator is the value of θ that maximizes $LL(\theta)$.

In simulated maximum likelihood, $\check{P}_n(\theta)$ is the simulated approximation to $P_n(\theta)$. The simulated log-likelihood function is:

$$SLL(\theta) = \sum_n \ln \check{P}_n(\theta) \tag{49}$$

The maximum simulated likelihood (MSL) estimator is the value of θ that maximizes $SLL(\theta)$.

It turns out that all of the simulators that we have looked at are unbiased for the true probability, P_n. However, because the log operation in the log-likelihood function is a non-linear transformation, $\ln\check{P}_n(\theta)$ is not unbiased for $\ln P_n(\theta)$ even though $\check{P}_n(\theta)$ is unbiased for $P_n(\theta)$. The bias in the simulator of $\ln\check{P}_n(\theta)$ translates into bias for the MSL estimator. Fortunately, this bias declines as we use more draws in the simulation.

Train (2007, 242) describes the asymptotic properties of the MSL estimator. He shows that if R is fixed, then MSL is inconsistent. If R rises with N, then MSL is consistent. And if R rises faster than $\sqrt{N}$, then MSL is asymptotically equivalent to ML.

3.5 Estimation

To estimate multinomial probit models in STATA 10, you use the ASMPROBIT command. Let's take a look at exactly how this works with the Dutch election data that we have been using. Recall what the data look like.

```
. list respid vote choice abortion nuclear incomedifference in
1/12, sepby(respid)

     +-------------------------------------------------------+
     | respid   vote   choice   abortion   nuclear   income~e |
     |-------------------------------------------------------|
  1. |      1      1        1          2         1          1 |
  2. |      1      2        0          5         4          3 |
  3. |      1      3        0          3         3          6 |
  4. |      1      4        0          3         3          2 |
     |-------------------------------------------------------|
  5. |      2      1        0          1         2          0 |
  6. |      2      2        1          1         0          2 |
  7. |      2      3        0          1         4          1 |
  8. |      2      4        0          1         2          1 |
     |-------------------------------------------------------|
  9. |      3      1        0          0         1          2 |
 10. |      3      2        0          3         3          0 |
 11. |      3      3        0          0         3          3 |
 12. |      3      4        1          0         2          0 |
     +-------------------------------------------------------+
```

3.5.1 Multinomial Probit with IID Errors

Imagine that we start by estimating a pure conditional logit model that includes alternative-specific constants. I created the alternative-specific constants using the dummy variable trick that we used last week; the omitted or base category is party 1 or the Pvda.

```
. clogit choice vot2 vot3 vot4 abortion nuclear incomedifference, group(respid) nolog

Conditional (fixed-effects) logistic regression   Number of obs   =       3508
                                                  LR chi2(6)      =     832.04
                                                  Prob > chi2     =     0.0000
Log likelihood =  -808.1245                       Pseudo R2       =     0.3398

------------------------------------------------------------------------------
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
        vot2 |   .5532445   .1106222     5.00   0.000     .3364291    .7700599
        vot3 |  -.1792512   .1245497    -1.44   0.150    -.4233641    .0648617
        vot4 |  -1.001535   .1160065    -8.63   0.000    -1.228903   -.7741659
    abortion |  -.3647413   .0329977   -11.05   0.000    -.4294156    -.300067
     nuclear |  -.2284208    .033909    -6.74   0.000    -.2948812   -.1619604
incomediff~e |   -.532328   .0364852   -14.59   0.000    -.6038377   -.4608183
------------------------------------------------------------------------------
```

To fit the corresponding multinomial probit model, you would type:

```
. asmprobit choice abortion nuclear incomedifference,
     case(respid) alternatives(vote) basealternative(1)
     correlation(independent) stddev(homoskedastic) nolog;

Alternative-specific multinomial probit        Number of obs      =       2944
Case variable: respid                          Number of cases    =        736

Alternative variable: vote                     Alts per case: min =          4
                                                              avg =        4.0
                                                              max =          4
Integration sequence:      Hammersley
Integration points:               200             Wald chi2(3)    =     361.06
Log simulated-likelihood =  -677.5638             Prob > chi2     =     0.0000

------------------------------------------------------------------------------
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
vote         |
    abortion |  -.2679146   .0260151   -10.30   0.000    -.3189032    -.216926
     nuclear |   -.172252   .0264295    -6.52   0.000    -.2240529   -.1204512
incomediff~e |  -.3542799   .0279023   -12.70   0.000    -.4089673   -.2995924
-------------+----------------------------------------------------------------
vote1        |  (base alternative)
-------------+----------------------------------------------------------------
vote2        |
       _cons |   .4293026   .0908491     4.73   0.000     .2512417    .6073636
-------------+----------------------------------------------------------------
vote3        |
       _cons |  -.2500803   .1006422    -2.48   0.013    -.4473353   -.0528253
-------------+----------------------------------------------------------------
vote4        |
       _cons |  -.7916313   .0895927    -8.84   0.000    -.9672298   -.6160327
------------------------------------------------------------------------------
(vote=1 is the alternative normalizing location)
(vote=2 is the alternative normalizing scale)
```

The variable indicating which records belong to a given case are now shown in CASE() rather than GROUP(). The variable indicating the alternative corresponding to a given observation within a case is given by ALTERNATIVES(). For the CLOGIT command, the base alternative was implicitly specified by not listing an indicator variable for VOT1; with ASMPROBIT, we use the BASEALTERNATIVE() option. The last two options specify the correlation structure and it is these options that make this particular specification the equivalent of the clogit model. The option CORRELATION() indicates that the errors are uncorrelated as in the clogit model. The STDDEV() option restricts all the variances to be equal as in the clogit model. To see exactly what these last two options do, we can type:

```
. estat covariance
  +------------------------------------------------------------+
  |              |     vote1      vote2      vote3      vote4  |
  |--------------+---------------------------------------------|
  |        vote1 |         1                                   |
  |        vote2 |         0          1                        |
  |        vote3 |         0          0          1             |
  |        vote4 |         0          0          0          1  |
  +------------------------------------------------------------+
  . estat correlation
  +---------------------------------------------+
  |              |  vote1   vote2   vote3   vote4 |
  |--------------+------------------------------|
  |        vote1 |  1.000                       |
  |        vote2 |  0.000   1.000               |
  |        vote3 |  0.000   0.000   1.000       |
  |        vote4 |  0.000   0.000   0.000   1.000 |
  +---------------------------------------------+
```

You can see that the errors are uncorrelated with unit variance as you would expect. To see that the two models produce essentially the same outcomes, you can compare the predicted probabilities from the two models.

```
. pwcorr clmpr mnppr
             |    clmpr    mnppr
-------------+------------------
       clmpr |   1.0000
       mnppr |   0.9964   1.0000
```

As you can see, there is a close match.[13]

3.5.2 Multinomial Probit with Correlated Errors – Full Covariance Matrix

Now let's look at the probit model without IID errors. The main reason to use ASMPROBIT rather than CLOGIT is to allow for correlated errors. Let's start by estimating the full covariance matrix version.

In the output below, INTEGRATION SEQUENCE: HAMMERSLEY indicates that a Hammersley sequence was used in the simulation process. INTEGRATION POINTS: 200 indicates how many draws were used to compute the integrals used for estimation. As we noted earlier, the default in STATA is to estimate the full covariance matrix. This involves using the differenced covariance parameterization outlined earlier so that the covariance matrix for the model in this situation with four alternatives is 3×3; there are two free variances to estimate and three correlations. The GHK simulator that is used in STATA involves using what is called a Choleski-factored variance-covariance. This is quite complicated, but the information that is

[13] However, it is worth noting that the correlations are not as high as they typically are between MPROBIT and MLOGIT. Moreover, there is a rather big difference in the log-likelihoods for the two models. Long and Freese (2006, 322) argue that noticeably different log-likelihoods can occur when the models are misspecified; they should not be very different if the models are well-specified.

produced at the bottom of the table refers to elements of the Cholesky matrix. For more information about this, see Train (2007, 127-137) and the STATA manuals. Although the transformed (Choleski-factored) parameters of the difference covariance parameterization at the bottom of the table are difficult to interpret, you can view them untransformed ($\tilde{\Omega}_1^*$) by using the ESTAT COVARIANCE command. You can also see the correlation between the normalized error differences by using the ESTAT CORRELATION command.

```
. asmprobit choice abortion nuclear incomedifference,
            case(respid) alternatives(vote) basealternative(1)

Alternative-specific multinomial probit        Number of obs      =       2944
Case variable: respid                          Number of cases    =        736

Alternative variable: vote                     Alts per case: min =          4
                                                              avg =        4.0
                                                              max =          4
Integration sequence:      Hammersley
Integration points:               200             Wald chi2(3)    =     147.99
Log simulated-likelihood = -668.02639             Prob > chi2     =     0.0000

------------------------------------------------------------------------------
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
vote         |
    abortion |  -.2525003   .0262676    -9.61   0.000    -.3039839   -.2010168
     nuclear |  -.1383659    .024649    -5.61   0.000     -.186677   -.0900548
incomediff~e |  -.2959191   .0320756    -9.23   0.000    -.3587862    -.233052
-------------+----------------------------------------------------------------
vote1        |  (base alternative)
-------------+----------------------------------------------------------------
vote2        |
       _cons |   .3498381   .0993009     3.52   0.000      .155212    .5444642
-------------+----------------------------------------------------------------
vote3        |
       _cons |  -.0742756   .1208308    -0.61   0.539    -.3110996    .1625484
-------------+----------------------------------------------------------------
vote4        |
       _cons |  -.9300537   .2494794    -3.73   0.000    -1.419024    -.441083
-------------+----------------------------------------------------------------
     /lnl2_2 |  -.3429903   .1703436    -2.01   0.044    -.6768576   -.0091229
     /lnl3_3 |   .1438355   .2659565     0.54   0.589    -.3774296    .6651006
-------------+----------------------------------------------------------------
       /l2_1 |   .4397972   .1201775     3.66   0.000     .2042536    .6753408
       /l3_1 |   .7781924   .2391611     3.25   0.001     .3094453     1.24694
       /l3_2 |  -.4236947   .4578219    -0.93   0.355    -1.321009    .4736197
------------------------------------------------------------------------------
(vote=1 is the alternative normalizing location)
(vote=2 is the alternative normalizing scale)


. estat covariance
  +------------------------------------------------+
  |              |     vote2      vote3      vote4 |
  |--------------+---------------------------------|
  |        vote2 |         2                       |
  |        vote3 |  .6219671   .6970178            |
  |        vote4 |   1.10053    .041574   2.118419 |
  +------------------------------------------------+
Note: covariances are for alternatives differenced with vote1
```

```
. estat correlation
  +------------------------------------+
  |              |  vote2   vote3   vote4 |
  |--------------+-------------------------|
  |        vote2 |  1.000                  |
  |        vote3 |  0.527   1.000          |
  |        vote4 |  0.535   0.034   1.000 |
  +------------------------------------+
Note: correlations are for alternatives differenced with vote1
```

The note at the bottom of the ESTAT COVARIANCE output indicates that this is the covariance matrix for the differences in errors relative to alternative VOTE1 (vote for Pvda): $(\epsilon_{n2} - \epsilon_{n1})$, $(\epsilon_{n3} - \epsilon_{n1})$, and $(\epsilon_{n4} - \epsilon_{n1})$. The variance for $(\epsilon_{n2} - \epsilon_{n1})$ is fixed, as we noted earlier for STATA, at 2 to identify the model. The five remaining elements are estimated. As I noted earlier, it might be tempting to use the information from ESTAT COVARIANCE or ESTAT CORRELATION to describe the correlations among the errors for the utilities. For example, you might *incorrectly* assume that the positive errors for vote3 are associated with positive errors for vote2. This is incorrect, though, since we are estimating these values after the constraints have been imposed. In other words, 0.527 is the estimated correlation between $(\epsilon_{n2} - \epsilon_{n1})$ and $(\epsilon_{n3} - \epsilon_{n1})$, which is unlikely to be of much substantive interest.

3.5.3 Multinomial Probit with Correlated Errors – Restricted Covariance Matrix

By default, ASMPROBIT estimates the full covariance matrix of *normalized error differences*, $\tilde{\Omega}_1^*$. You can, though, use the STRUCTURAL option to estimate the structural covariance matrix of the *undifferenced errors*, Ω. As you will remember, not all of the parameters of Ω are identified and so more identification restrictions are required. This forces us to estimate a restricted covariance matrix. As we saw earlier, in a model with J alternatives, there are $J(J+1)/2$ distinct elements in the covariance matrix. However, the data only identify $[(J-1)J/2] - 1$ of these elements. As a result, to estimate a structural covariance matrix, you must impose at least $J+1$ restrictions. By default, ASMPROBIT ..., STRUCTURAL sets two error variances to 1, which provides two restrictions. $J-1$ additional restrictions have to be imposed by setting the error covariances between the base alternative and the other alternatives to 0. For example, in a situation with four alternatives, you would start with the following:

$$\Omega = \begin{bmatrix} \sigma_{11} & \cdot & \cdot & \\ \sigma_{21} & \sigma_{22} & \cdot & \cdot \\ \sigma_{31} & \sigma_{32} & \sigma_{33} & \cdot \\ \sigma_{41} & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{bmatrix} \tag{50}$$

We might want to restrict the covariance matrix so that it looks like the following:

$$\Omega = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot \\ 0 & \sigma_{32} & \sigma_{33} & \cdot \\ 0 & \sigma_{42} & \sigma_{43} & \sigma_{44} \end{bmatrix} \tag{51}$$

As you can see, we have set two of the error variances to 1 and then we have made 3 $(J-1)$ other restrictions by setting three of the covariances for alternative 1 to 0. To do this with the STRUCTURAL option, we could

(i) use BASE(1) to specify that $\sigma_{11} = 1$ and $\sigma_{j1} = 0$ and (ii) use SCALE(2) to specify that $\sigma_{22} = 1$. We could (theoretically) estimate this model for our Dutch elections data by typing:

```
asmprobit choice abortion nuclear incomedifference,
        case(respid) alternatives(vote) base(1) scale(2) structural;
```

In fact, simply using the STRUCTURAL command would also work since the default is BASE(1) and SCALE(2) i.e.

```
asmprobit choice abortion nuclear incomedifference,
        case(respid) alternatives(vote) structural;
```

Unfortunately, these models do not converge for the Dutch election data that we have been looking at. One option is to add more restrictions.[14]

STATA actually allows you to choose different types of restricted covariance matrices. For example, a more restrictive option would be to go with STATA's EXCHANGEABLE option. This option specifies the following *correlation* matrix between the errors for a four-alternative scenario:

$$\text{Correlation Matrix} = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot \\ 0 & \rho & 1 & \cdot \\ 0 & \rho & \rho & 1 \end{bmatrix} \tag{52}$$

In effect, this option is like the default structural setup but assumes that there is a common error correlation for each pair of alternatives. You can do a likelihood-ratio test where the null is that there is a common correlation. In this case, the LR test indicates that a common correlation is not a plausible hypothesis because the p-value is 0.01. You can get the correlation matrix and the covariance matrix of the errors by using ESTAT CORRELATION and ESTAT COVARIANCE.

[14] I'll have to play around with this some more and see if I can get some results for next time.

```
. asmprobit choice abortion nuclear incomedifference,
        case(respid) alternatives(vote)  correlation(exchangeable)
Alternative-specific multinomial probit        Number of obs      =       2944
Case variable: respid                          Number of cases    =        736
Alternative variable: vote                     Alts per case: min =          4
                                                              avg =        4.0
                                                              max =          4
Integration sequence:      Hammersley
Integration points:               200             Wald chi2(3)    =     204.57
Log simulated-likelihood = -672.48903             Prob > chi2     =     0.0000
------------------------------------------------------------------------------
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
vote         |
    abortion |  -.2667072   .0273705    -9.74   0.000    -.3203525    -.213062
     nuclear |  -.1604062   .0265506    -6.04   0.000    -.2124444   -.1083679
incomediff~e |  -.3356453   .0315326   -10.64   0.000    -.3974481   -.2738425
-------------+----------------------------------------------------------------
vote1        |  (base alternative)
-------------+----------------------------------------------------------------
vote2        |
       _cons |   .3919668   .1044861     3.75   0.000     .1871778    .5967558
-------------+----------------------------------------------------------------
vote3        |
       _cons |  -.0441727   .1317075    -0.34   0.737    -.3023147    .2139692
-------------+----------------------------------------------------------------
vote4        |
       _cons |  -1.152429   .2734464    -4.21   0.000    -1.688374   -.6164838
-------------+----------------------------------------------------------------

. lrtest full .
Likelihood-ratio test                                 LR chi2(2)  =      8.93
(Assumption: . nested in full)                        Prob > chi2 =    0.0115


. estat correlation
  +-------------------------------------------------+
  |              |  vote1   vote2   vote3   vote4 |
  |--------------+----------------------------------|
  |        vote1 |  1.000                           |
  |        vote2 |  0.000   1.000                   |
  |        vote3 |  0.000  -0.198   1.000           |
  |        vote4 |  0.000  -0.198  -0.198   1.000 |
  +-------------------------------------------------+

. estat covariance
  +------------------------------------------------------------+
  |              |     vote1      vote2      vote3      vote4 |
  |--------------+---------------------------------------------|
  |        vote1 |         1                                   |
  |        vote2 |         0          1                        |
  |        vote3 |         0  -.0991577   .2513414             |
  |        vote4 |         0  -.2655532  -.1331323    1.80266 |
  +------------------------------------------------------------+
```

You can allow for different correlations across pairs of alternatives by using the UNSTRUCTURED option.

$$\text{Correlation Matrix} = \begin{bmatrix} 1 & \cdot & \cdot & \cdot \\ 0 & 1 & \cdot & \cdot \\ 0 & \rho_{32} & 1 & \cdot \\ 0 & \rho_{42} & \rho_{43} & 1 \end{bmatrix} \tag{53}$$

```
. asmprobit choice abortion nuclear incomedifference,
    case(respid) alternatives(vote)  correlation(unstructured)

Alternative-specific multinomial probit        Number of obs      =       2944
Case variable: respid                          Number of cases    =        736

Alternative variable: vote                     Alts per case: min =          4
                                                              avg =        4.0
                                                              max =          4
Integration sequence:      Hammersley
Integration points:               200             Wald chi2(3)    =     147.99
Log simulated-likelihood = -668.02639             Prob > chi2     =     0.0000

------------------------------------------------------------------------------
      choice |      Coef.   Std. Err.      z    P>|z|     [95% Conf. Interval]
-------------+----------------------------------------------------------------
vote         |
    abortion |  -.2525003   .0262676    -9.61   0.000    -.3039839   -.2010168
     nuclear |  -.1383659    .024649    -5.61   0.000     -.186677   -.0900548
incomediff~e |  -.2959191   .0320756    -9.23   0.000    -.3587862    -.233052
-------------+----------------------------------------------------------------
vote1        |  (base alternative)
-------------+----------------------------------------------------------------
vote2        |
       _cons |   .3498381   .0993009     3.52   0.000      .155212    .5444642
-------------+----------------------------------------------------------------
vote3        |
       _cons |  -.0742756   .1208308    -0.61   0.539    -.3110996    .1625484
-------------+----------------------------------------------------------------
vote4        |
       _cons |  -.9300537   .2494794    -3.73   0.000    -1.419024    -.441083
-------------+----------------------------------------------------------------

. estat covariance

  +-------------------------------------------------+
  |              |     vote2      vote3      vote4 |
  |--------------+----------------------------------|
  |        vote2 |         2                        |
  |        vote3 |  .6219671   .6970178             |
  |        vote4 |   1.10053    .041574   2.118419 |
  +-------------------------------------------------+
Note: covariances are for alternatives differenced with vote1

. estat correlation

  +---------------------------------------+
  |              |  vote2   vote3   vote4 |
  |--------------+------------------------|
  |        vote2 |  1.000                 |
  |        vote3 |  0.527   1.000         |
  |        vote4 |  0.535   0.034   1.000 |
  +---------------------------------------+
Note: correlations are for alternatives differenced with vote1
```

You can actually choose more specific correlation patterns. For example, you could specify fixed values for some correlations and free parameters for others by using the CORRELATION(FIXED *matname*) option (see the STATA manuals).

However you specify the structure of the covariance matrix when using the structural option, you should verify that all the parameters of the resulting model are actually identified. We already saw how to check

this from earlier – you demonstrate that the restricted elements in the covariance matrix of the errors can be calculated from the elements of the normalized covariance matrix of error differences. The bottom line is that you need to be careful in all of this.

3.6 Interpretation

When interpreting your results, you can look at a summary of the alternatives and frequencies in the estimation sample. After estimating your model, you can type the following:

```
. estat alternatives
   Alternatives summary for vote
  +-------------------------------------------------------------------------+
  |Covariance      Alt              Alt   |    Cases   Frequency    Percent |
  |  index       value            label   |   present   selected   selected |
  |-----------------------------------------+---------------------------------|
  |     1            1                1   |       736        273      37.09 |
  |     2            2                2   |       736        247      33.56 |
  |     3            3                3   |       736        116      15.76 |
  |     4            4                4   |       736        100      13.59 |
  +-------------------------------------------------------------------------+
```

You can get STATA to give you the predicted probability that each alternative is chosen by the individuals in your data set using the PREDICT command.

```
. asmprobit choice abortion nuclear incomedifference,
    case(respid) alternatives(vote)  correlation(unstructured)

. predict prob
(option pr assumed; Pr(vote))

. list respid vote choice prob in 1/8
     +-----------------------------------+
     | respid   vote   choice       prob |
     |-----------------------------------|
  1. |      1      1        1   .7774967 |
  2. |      1      2        0   .1344112 |
  3. |      1      3        0   .0022694 |
  4. |      1      4        0   .0857648 |
  5. |      2      1        0    .408968 |
     |-----------------------------------|
  6. |      2      2        1    .440839 |
  7. |      2      3        0   .0689605 |
  8. |      2      4        0   .0808914 |
     +-----------------------------------+
```

You can also use the ASPRVALUE command from SPOST. For example, if you wanted to know the probability of voting for each of the parties when all of the alternative specific variables shared the same (mean) value, then you would have:

```
. asprvalue
asmprobit: Predictions for choice

        prob
1  .26519328
2  .46764544
3  .17391652
4  .09278571

alternative-specific variables
                          1          2          3          4
        abortion  1.5285326  1.5285326  1.5285326  1.5285326
         nuclear   1.763587   1.763587   1.763587   1.763587
incomedifference  1.8695652  1.8695652  1.8695652  1.8695652
```

In other words, these are the probabilities of voting for parties 1-4 when all four parties are 1.53 units away from the voter on abortion, 1.76 units away on the nuclear dimension, and 1.87 units away on the income dimension.

You might also be interested in computing predicted probabilities for the situation where each alternative party has its average values on the alternative specific variables:

```
. asprvalue, rest(asmean)
asmprobit: Predictions for choice
        prob
1  .39734545
2  .35238993
3  .09606214
4  .15357095

alternative-specific variables
                          1          2          3          4
        abortion  1.5285326  2.9605978     1.8125  1.5638587
         nuclear   1.763587  2.4021739  2.8342391  1.6657609
incomedifference  1.8695652  1.9782609  2.7486413  1.5067935
```

As you can see, the difference is simply that we have the average distance each party is away from the voters on each dimension i.e. they differ across alternatives.

You can set alternative specific variables to specific values too. For example, what happens to the probability of choosing each party if we move each party 0.5 units further away on the abortion dimension?

```
. quietly asprvalue, x(abortion=1.528536 2.9605978 1.8125 1.5638587) rest(asmean) save
. asprvalue,  x(abortion=2.028536 3.4605978 2.3125 2.0638587) rest(asmean) diff

asmprobit: Predictions for choice
      Current       Saved        Diff
1   .39734513   .39734516  -2.980e-08
2   .35239008   .35239008           0
3   .09606226   .09606225   7.451e-09
4   .15357104   .15357102   1.490e-08
```

```
alternative-specific variables
                              1          2          3          4
        Current:abortion   2.028536  3.4605978     2.3125  2.0638587
         Current:nuclear   1.763587  2.4021739  2.8342391  1.6657609
Current:incomedifference  1.8695652  1.9782609  2.7486413  1.5067935
          Saved:abortion   1.528536  2.9605978     1.8125  1.5638587
           Saved:nuclear   1.763587  2.4021739  2.8342391  1.6657609
  Saved:incomedifference  1.8695652  1.9782609  2.7486413  1.5067935
            Dif:abortion         .5         .5         .5         .5
             Dif:nuclear          0          0          0          0
    Dif:incomedifference          0          0          0          0
```

4 Mixed Logit Model

The mixed logit model avoids the three limitations that we outlined earlier for logit models by allowing for random taste variation, unrestricted substitution patterns, and correlation in unobserved factors over time. Unlike the probit model we just examined, the mixed logit model is not restricted to normal distributions. McFadden and Train (2000) have shown that the mixed logit is highly flexible and can approximate any random utility model.

Mixed logit models are defined based on the functional form chosen for the choice probabilities. In effect, mixed logit probabilities are the integrals of standard logit probabilities over a density of parameters. In other words, mixed logit probabilities have the following generic form:

$$P_{ni} = \int L_{ni}(\beta) f(\beta) d\beta \tag{54}$$

where $L_{ni}(\beta)$ is the logit probability evaluated at parameters β i.e.

$$L_{ni}(\beta) = \frac{e^{V_{ni}(\beta)}}{\sum_{j=1}^{J} e^{V_{nj}(\beta)}} \tag{55}$$

$V_{ni}(\beta)$ is the observed portion of utility, which depends on β. If utility is linear in β, the usual setup, then we have $V_{ni}(\beta) = x_{ni}\beta$. This leads to the usual mixed logit probability:

$$P_{ni} = \int \left(\frac{e^{x_{ni}\beta}}{\sum_j e^{x_{nj}\beta}} \right) f(\beta) d\beta \tag{56}$$

As you can see, the mixed logit probability is just a weighted average of the standard logit formula evaluated at different values of β, with the weights given by the density $f(\beta)$. When something is a weighted average of several functions it is *mixed function* and the density that provides the weights, $f(\beta)$, is called the *mixing distribution*.

It should be obvious that the standard logit model is a special case of the mixed logit where $f(\beta)$ is degenerate at fixed parameters b: $f(\beta) = 1$ for $\beta = b$ and 0 for $\beta \neq b$. Note that the mixing distribution $f(\beta)$ can take on any form. For example, you could specify $f(\beta)$ to be discrete, with β taking on a finite set of distinct values. This type of setup is called a *latent class model* and looks like the following. Suppose β

takes on M possible values labeled $b_1, \ldots, b_M$, with probability s_m that $\beta = b$, then we have:

$$P_{ni} = \sum_{m=1}^{M} s_m \left(\frac{e^{x_{ni}\beta_m}}{\sum_j e^{x_{nj}\beta_m}} \right) \tag{57}$$

This type of setup might be useful if you think that there are M segments of a population with their own choice behavior or preferences.

In most applied cases, though, analysts have chosen a continuous distribution for $f(\beta)$. For example, you might let the density of β be normal with mean b and covariance W. The choice probability for this model would be:

$$P_{ni} = \int \frac{e^{x_{ni}\beta}}{\sum_j e^{x_{nj}\beta}} \phi(\beta|b, W) d\beta \tag{58}$$

where $\phi(\beta|b, W)$ is the normal density with mean b and covariance W. You can obviously choose other distributions such as the lognormal, uniform, triangular, gamma etc.. It is by choosing different distributions that you can approximate any utility-maximizing behavior.

Note that when you estimate a mixed logit model, you are estimating two sets of parameters. First, you have the parameters β that enter the logit formula; these parameters have density $f(\beta)$. Second, you have the set of parameters that describe this density. For example, with our normal example, b and W are parameters that describe the density $f(\beta)$; we typically want to estimate these as well. If we denote the parameters that describe the density of β as θ, then we have:

$$P_{ni} = \int L_{ni}(\beta) f(\beta|\theta) d\beta \tag{59}$$

which is a function of θ. We then integrate the β parameters out. Notice what is happening here. In effect, the β's are similar to the ϵ_{nj}'s from earlier in that both are random terms that are integrated out to obtain the choice probabilities. We end up estimating in this case b and W.

4.1 Random Coefficients Setup

The mixed logit model probabilities outlined above can be derived from a random utility model setup in a variety of ways: random coefficients and error components. Let's start with then random coefficients setup. Suppose that individual n is choosing among J alternatives. As before, the utility of individual n for alternative j is:

$$U_{nj} = x_{nj}\beta_n + \epsilon_{nj} \tag{60}$$

where β_n is a vector of coefficients for variables x_{nj} for person n that represent that person's tastes and ϵ_{nj} is a random term that is iid extreme value. The coefficients vary over individuals in the population according to the density $f(\beta)$. This density is a function of parameters θ that represent, say, the mean and covariance of the β's in the population. Note that this setup if exactly the same as a standard logit model setup except that the coefficients, β, vary over individuals rather than being fixed; this is the random coefficients bit.

Each individual n knows the value of his own β_n and ϵ_{nj}'s for all j and chooses alternative i if and only if $U_{ni} > U_{nj} \forall j \neq i$. In contrast, the analyst only observes x_{nj} and does not observe β_n or the ϵ_{nj}'s. If the analyst could observe β_n, then the choice probability would be a standard logit model since the ϵ_{nj}'s are iid extreme value. In other words, the choice probability *conditional* on β_n is:

$$L_{ni}(\beta) = \frac{e^{x_{ni}\beta_n}}{\sum_j e^{x_{nj}\beta_n}} \tag{61}$$

The problem is that the analyst does not know β_n and so can't condition on it. The unconditional choice probability is therefore the integral of L_{ni} over all possible values of β_n:

$$P_{ni} = \int \left(\frac{e^{x_{ni}\beta}}{\sum_j e^{x_{nj}\beta}} \right) f(\beta) d\beta \tag{62}$$

which is the mixed logit probability outlined earlier. Essentially, the analyst then specifies a distribution for the coefficients and estimates the parameters of that distribution.

Typically, analysts specify a normal distribution i.e. $\beta \sim N(b, W)$ with parameters b and W estimated. The log normal distribution $\ln\beta \sim N(b, W)$ might be used if the coefficient is known to have the same sign for every individual. For example, suppose we are interested in a spatial model of elections and we have a number of variables capturing the ideological distance from the parties to each voter on a number of different dimensions. A spatial model of elections would imply that as parties move further away from voters on any of these dimensions, then the voter should be less likely to choose that party. This would lead us to expect the coefficient on any 'distance' variable to be negative for all voters. As a result, we might specify a log normal distribution for the coefficients on these 'distance' variables (Glasgow 2001). Note that you can specify different distributions for different elements in β if you want. For example, you could specify a log normal distribution for some variables, a normal distribution for others, and a triangular distribution for others etc..

4.2 Error Components Setup

Now let's look at the error components setup. Remember that error components represent correlations among the utilities for different alternatives such that we have:

$$U_{nj} = x_{nj}\alpha + z_{nj}\eta_n + \epsilon_{nj} \tag{63}$$

where x_{nj} and z_{nj} are vectors of observed variables relating alternative j to individual n, α is a vector of fixed coefficients, and η is a vector of random terms with zero mean, and ϵ_{nj} is iid extreme value. The variables in z_{nj} are the error components that, along with ϵ_{nj}, determine the stochastic part of the utility. In other words, the unobserved or random component of utility is:

$$e_{nj} = z_{nj}\eta_n + \epsilon_{nj} \tag{64}$$

which can be correlated over alternatives depending on the variables in z_{nj}. Thus, our utility function is now:

$$U_{nj} = x_{nj}\alpha + e_{nj} \tag{65}$$

This model is essentially the same as a standard logit model if $z_{nj} = 0$ so that there is no correlation in utility over alternatives; as we saw earlier, it is this zero correlation that leads to the IIA assumption. In other words, the standard logit model is a special case of the error components mixed logit model. With nonzero error components, utility is correlated over alternatives: $\text{cov}(e_{ni}, e_{nj}) = E(z_{ni}\eta_n + \epsilon_{ni})(z_{nj}\eta_n + \epsilon_{nj}) = z'_{ni}Wz_{nj}$, where W is the covariance of η_n. Note that utility is correlated over alternatives even when the error components are independent.

It is possible to obtain various correlation and, hence, substitution patterns by choosing appropriate variables as error components. Train (2007, 143) illustrates how you can set up the mixed logit analog of a nested logit model. He states that you need to specify a dummy variable for each nest that equals 1 for each alternative in the nest and 0 for alternatives outside of the nest. With K non-overlapping nests, the error components would be:

$$z_{nj}\eta_n = \sum_{k=1}^{K} \eta_{nk} d_{jk} \tag{66}$$

where $d_{jk} = 1$ if j is in nest k and 0 otherwise. As Train notes, you can specify the error components to be independently normally distributed: η_{nk} iid $N(0, \sigma_k)$. The random quantity η_{nk} enters the utility of each alternative in nest k, inducing correlation across alternatives; it does not enter the utility of alternatives in nests other than k and so does not induce correlation across alternatives in different nests. The variance σ_k captures the magnitude of correlation and is, therefore, equivalent to the dissimilarity parameter λ_k that we saw when we looked at the nested logit model earlier.

4.3 Random Coefficients and Error Components Again

Random coefficient and error component specifications are formally equivalent. In the random coefficient setup, utility is $U_{nj} = x_{nj}\beta_n + \epsilon_{nj}$ with random β_n. As you will recall from the random coefficients probit model we saw earlier, it is possible to decompose the coefficients β_n into their mean α and their deviations η_n, so that $U_{nj} = x_{nj}\alpha + z_{nj}\eta_n + \epsilon_{nj}$, which has error components defined by $z_{nj} = x_{nj}$ i.e. the elements of x and z are identical. In contrast, under an error components setup, utility is $U_{nj} = x_{nj}\alpha + z_{nj}\eta_n + \epsilon_{nj}$, which is equivalent to a random-parameter model with fixed coefficients for x_{nj} and random coefficients with zero means for variables z_{nj}. If x_{nj} and z_{nj} overlap, then the coefficients of these variables vary randomly with mean α and the same distribution as η_n around their means; in this case, we have a mixture of a random coefficients and an error component setup.

4.4 Mixed Logit vs. Multinomial Probit

Compared to a mixed logit model, multinomial probit models have two properties that limit the types of situation that they can model (Glasgow 2001, 121-122). The first obvious restriction relates to the use of a normal distribution. Recall from earlier that random coefficient and error component models can both be written in the following format:

$$U_{nj} = x_{nj}\beta_j + (z_{nj}\eta_n + \epsilon_{nj}) \tag{67}$$

Probit models require that all of the terms in η_n be distributed normally. However, we know that there are many instances in which nonnormal distributions on error components or random coefficients are appropriate. Probit models cannot handle these situations but mixed logit models can.

The second restriction relates to the number of random terms that can be estimated with a probit model. We saw earlier that because only differences in utility matter and we have to set the scale of utility, only [(J-1)J/2]-1 elements of the covariance matrix Ω are identified. This means that probit models can estimate at most [(J-1)J/2]-1 random coefficients or error components. This is true regardless of the number of elements in z_{nj}. For example, in a model with 3 alternatives, I can have at most 2 random coefficients. However, it might be the case that you want to examine random taste variation in a greater number of coefficients. The only way to do this is with a mixed logit model; any number of elements can be included in the random term η.

4.5 Simulation

As with the probit models, we would estimate a mixed logit model using simulated maximum likelihood. The choice probabilities are:

$$P_{ni} = \int L_{ni}(\beta) f(\beta|\theta) d\beta \tag{68}$$

where

$$L_{ni}(\beta) = \frac{e^{x_{ni}\beta}}{\sum_j e^{x_{nj}\beta}} \tag{69}$$

The analyst specifies $f(\cdot)$ and wants to estimate parameters θ. These probabilities are approximated through simulation for any given value of θ: (i) draw a value of β from $f(\beta|\theta)$ and label it β^r where $r = 1$ referring to the first draw; (ii) calculate the logit formula $L_{ni}(\beta^r)$ with this draw; (iii) repeat steps (i) and (ii) many times, and then average the results. This average is the simulated probability:

$$\check{P}_{ni} = \frac{1}{R}\sum_{r=1}^{R} L_{ni}(\beta^r) \tag{70}$$

where R is the number of draws. These simulated probabilities are then put into the log-likelihood function to get a simulated log-likelihood function:

$$\text{simulated } \ln\mathcal{L} = SLL = \sum_n \sum_j d_{nj} \ln \check{P}_{nj} \tag{71}$$

where $d_{nj} = 1$ if n chose j and 0 otherwise. The maximum simulated likelihood estimator (MSLE) is the value of θ that maximizes SLL.

4.6 Train's Case Study

Train (2007, 151-154) provides the following random coefficients mixed logit example. He is interested in the choices that anglers make about where they fish. The utility function is:

$$U_{nj} = x_{nj}\beta_n + \epsilon_{nj} \tag{72}$$

with coefficients β_n varying over anglers. The results of his analysis are shown in Table 2. Train expects the coefficients for FISH STOCK, AESTHETICS, and TOTAL COST to have the same sign for all anglers but that their magnitude will vary across anglers. As a result, these coefficients are given independent lognormal distributions. The mean and standard deviation of the log of the coefficient are estimated, and the mean and standard deviation of the coefficient itself are calculated from these estimates. Train expects that the coefficients on GUIDE LISTS AS MAJOR, CAMPGROUNDS, ACCESS AREAS, and RESTRICTED SPECIES can take on either positive or negative signs; for example, some anglers might like having campgrounds but others might dislike having them. As a result, each of these coefficients is given an independent normal distribution with mean and standard deviation that are estimated. Train expects that the coefficient on the log of size is fixed, thereby allowing for the possibility that the probability of visiting a larger site is higher than that for a smaller site; Train does not allow the coefficient for this variable to vary over anglers (though you might).

Table 2: Mixed Logit Model of River Fishing Site Choice

Variable	Parameter	Value	Std. Error
Fish stock	Mean of ln(coefficient)	-2.876	0.6066
	Std. dev. of ln(coefficient)	1.016	0.2469
Aesthetics	Mean of ln(coefficient)	-0.794	0.2287
	Std. dev. of ln(coefficient)	0.849	0.1382
Trip cost	Mean of ln(coefficient)	-2.402	0.0631
	Std. dev. of ln(coefficient)	0.801	0.0781
Guide lists as major	Mean coefficient	1.018	0.2887
	Std. dev. of coefficient	2.195	0.3518
Campgrounds	Mean coefficient	0.116	0.3233
	Std. dev. of coefficient	1.655	0.4350
Access areas	Mean coefficient	-0.950	0.3610
	Std. dev. of coefficient	1.888	0.3511
Restricted species	Mean coefficient	-0.499	0.1310
	Std. dev. of coefficient	0.899	0.1640
Log(size)	Mean coefficient	0.984	0.1077
SLL at covergence		-1932.33	

What can you infer from the results? Let's start by looking at the estimated mean coefficients. The signs indicate the direction of the average effect of these variables on the dependent variable. The standard deviation of each random coefficient is highly significant, indicating that these coefficients really do vary in the population. Now let's look at the normally distributed coefficients. The estimated means and standard deviations of these coefficients provide information on the share of the population that places a positive value on a site attribute and the share that places a negative value. Look at the information on GUIDES LISTS AS MAJOR. With an estimated mean of 1.018 and a standard deviation of 2.195, we infer that 68% of the distribution is above 0 and 32% is below 0 i.e. roughly two-thirds of anglers see being listed as a major site as a positive factor and one third sees it as a negative factor. Now let's look at the lognormal coefficients. Coefficient β^k follows a lognormal if the log of β^k is normally distributed. Train parameterizes the lognormal distribution in terms of the underlying normal i.e. he estimates m and s that represent the mean and variance of the log of the coefficient: $\ln\beta^k \sim N(m, s)$. You can then get the mean and variance of β^k from the estimates of m and s. The median is e^m, the mean is $e^{m+s/2}$, and the variance is $e^{2m+s}[e^s - 1]$. The estimates from Table 2 indicate that the coefficients of FISH STOCK, AESTHETICS, and TOTAL COST have the following median, mean, and standard deviation:

Variable	Median	Mean	Std. Dev.
Fish stock	0.0563	0.0944	0.1270
Aesthetics	0.4519	0.6482	0.6665
Trip cost	0.0906	0.1249	0.1185

As you can see, the mixed logit model provides much more information than the standard logit models examined earlier. Train (2007, 262-284) provides insights into how to gain even more information from mixed logit models.

Train's case study that we just examined used a random coefficients setup. To see an example using an error components setup, see Glasgow (2001, 123-126).

References

Bhat, C. 2001. "Quasi-Random Maximum Simulated Likelihood Estimation of the Mixed Multinomial Logit Model." *Transportation Research* 35:677–693.

Glasgow, Garrett. 2001. "Mixed Logit Models for Multiparty Elections." *Political Analysis* 9:116–136.

Greene, William. 2003. *Econometric Analysis*. New Jersey: Prentice Hall.

Haaijer, M., M. Wedel, M. Vriens & T. Wansbeek. 1998. "Utility Covariances and Context Effects in Conjoint MNP Models." *Marketing Science* 17:236–252.

Hammersley, J. & K. Morton. 1956. "A New Monte Carlo Technique: Antithetic Variates." *Proceedings of the Cambridge Philosophical Society* 52:449–474.

Hausman, J. & D. Wise. 1978. "A Conditional Probit Model for Qualitative Choice: Discrete Decisions Recognizing Interdependence and Heterogeneous Preferences." *Econometrica* 48:403–429.

Heiss, F. 2002. "Structural Choice Analysis with Nested Logit Models." *STATA Journal* 2:227–252.

Henscher, D. A., J. M. Rose & W. H. Greene. 2005. *Applied Choice Analysis: A Primer*. New York: Cambridge University Press.

Hess, Stephanie, Michel Bierlaire & John W. Polak. 2004. "Development and Application of a Mixed Cross-Nested Logit Model." Manuscript.

Long, J. Scott & Jeremy Freese. 2006. *Regression Models for Categorical Dependent Variables Using STATA*. Texas: STATA Corporation.

McFadden, D. & K. Train. 2000. "Mixed MNL Models of Discrete Response." *Journal of Applied Econometrics* 15:447–470.

Rivers, Doug. 1988. "Heterogeneity in Models of Electoral Choice." *American Journal of Political Science* 32:737–757.

Train, Kenneth E. 2007. *Discrete Choice Models with Simulation*. New York: Cambridge University Press.

Vovsha, Peter. 1997. "Application of a Cross-Nested Logit Model to Mode Choice in Tel Aviv, Israel, Metropolitan Area." *Transportation Research Record* 1607:6–15.

Wen, Chieh-Hua & Frank S. Koppelman. 2001. "The Generalized Nested Logit." *Transportation Research Part B* 35:627–641.