SAS/ETS® 13.2 User's Guide

The MDC Procedure

Chapter 18
The MDC Procedure

Contents

Overview: MDC Procedure

The MDC (multinomial discrete choice) procedure analyzes models in which the choice set consists of multiple alternatives. This procedure supports conditional logit, mixed logit, heteroscedastic extreme value, nested logit, and multinomial probit models. The MDC procedure uses the maximum likelihood (ML) or simulated maximum likelihood method for model estimation. The term *multinomial logit* is often used in the econometrics literature to refer to the *conditional logit* model of McFadden (1974). Here, the term *conditional logit* refers to McFadden's conditional logit model, and the term *multinomial logit* refers to a model that differs slightly. Early applications of the multinomial logit model in the econometrics literature are provided by Schmidt and Strauss (1975); Theil (1969). The main difference between McFadden's conditional logit model and the multinomial logit model is that the multinomial logit model makes the choice probabilities depend on the characteristics of the individuals only, whereas the conditional logit model considers the effects of choice attributes on choice probabilities as well.

Unordered multiple choices are observed in many settings in different areas of application. For example, choices of housing location, occupation, political party af liation, type of automobile, and mode of transportation are all unordered multiple choices. Economics and psychology models often explain observed choices by using the *random utility* function. The utility of a speci c choice can be interpreted as the relative pleasure or happiness that the decision maker derives from that choice with respect to other alternatives in a nite choice set. It is assumed that the individual chooses the alternative for which the associated utility is highest. However, the utilities are not known to the analyst with certainty and are therefore treated by the analyst as random variables. When the utility function contains a random component, the individual choice behavior becomes a probabilistic process.

The random utility function of individual i for choice j can be decomposed into deterministic and stochastic components

$$U_{ij} = V_{ij} + \varepsilon_{ij}$$

where V_{ij} is a deterministic utility function, assumed to be linear in the explanatory variables, ε_{ij} is an unobserved random variable that captures the factors that affect utility that are not included in V_{ij}. Different assumptions on the distribution of the errors, ε_{ij}, give rise to different classes of models.

The features of discrete choice models available in the MDC procedure are summarized in Table 18.1.

Table 18.1 Summary of Models Supported by PROC MDC

Model Type	Utility Function	Distribution of ε_{ij}
Conditional logit	$U_{ij} = x_{ij}'\beta + \varepsilon_{ij}$	IEV, independent and identical
HEV	$U_{ij} = x_{ij}'\beta + \varepsilon_{ij}$	HEV, independent and nonidentical
Nested logit	$U_{ij} = x_{ij}'\beta + \varepsilon_{ij}$	GEV, correlated and identical
Mixed logit	$U_{ij} = x_{ij}'\beta + z_{ij}'\gamma_i + \varepsilon_{ij}$	IEV, independent and identical
Multinomial probit	$U_{ij} = x_{ij}'\beta + \varepsilon_{ij}$	MVN, correlated and nonidentical

IEV stands for type I extreme-value (or Gumbel) distribution with the probability density function and the cumulative distribution function of the random error given by $f(\varepsilon_{ij}) = \exp(-\varepsilon_{ij})\exp(-\exp(-\varepsilon_{ij}))$ and $F(\varepsilon_{ij}) = \exp(-\exp(-\varepsilon_{ij}))$. HEV stands for heteroscedastic extreme-value distribution with the probability density function and the cumulative distribution function of the random error given by $f(\varepsilon_{ij}) = \frac{1}{\sigma_j}\exp(-\frac{\varepsilon_{ij}}{\sigma_j})\exp(-\exp(-\frac{\varepsilon_{ij}}{\sigma_j}))$ and $F(\varepsilon_{ij}) = \exp(-\exp(-\frac{\varepsilon_{ij}}{\sigma_j}))$, where σ_j is a scale parameter for the random component of the jth alternative. GEV stands for generalized extreme-value distribution. MVN represents multivariate normal distribution; and γ_i is an error component. See the "Mixed Logit Model" on page 1026 section for more information about γ_i.

Getting Started: MDC Procedure

Conditional Logit: Estimation and Prediction

The MDC procedure is similar in use to the other regression model procedures in the SAS System. However, the MDC procedure requires identification and choice variables. For example, consider a random utility function

$$U_{ij} = x_{1;ij}\beta_1 + x_{2;ij}\beta_2 + \varepsilon_{ij} \quad j = 1;\ldots;3$$

where the cumulative distribution function of the stochastic component is a Type I extreme-value, $F(\varepsilon_{ij}) = \exp(-\exp(-\varepsilon_{ij}))$. You can estimate this conditional logit model with the following statements:

```
proc mdc;
    model decision = x1 x2 / type=clogit
        choice=(mode 1 2 3);
    id pid;
run;
```

Note that the MDC procedure, unlike other regression procedures, does not include the intercept term automatically. The dependent variable decision takes the value 1 when a specific alternative is chosen; otherwise, it takes the value 0. Each individual is allowed to choose one and only one of the possible alternatives. In other words, the variable decision takes the value 1 one time only for each individual. If each individual has three elements (1, 2, and 3) in the choice set, the NCHOICE=3 option can be specified instead of CHOICE=(mode 1 2 3).

Consider the following trinomial data from Daganzo (1979). The original data set (origdata) contain travel time (ttime1–ttime3) and choice (choice) variables. The variables ttime1–ttime3 are the travel times for three different modes of transportation, and choice indicates which one of the three modes is chosen. The choice variable must have integer values.

```
data origdata;
   input ttime1 ttime2 ttime3 choice @@;
datalines;
16.481  16.196  23.89   2  15.123  11.373  14.182  2
19.469  8.822   20.819  2  18.847  15.649  21.28   2
12.578  10.671  18.335  2  11.513  20.582  27.838  1
10.651  15.537  17.418  1  8.359   15.675  21.05   1

   ... more lines ...
```

A new data set (newdata) is created because PROC MDC requires that each individual decision maker has one case for each alternative in his choice set. Note that the ID statement is required for all MDC models. In the following example, there are two public transportation modes, 1 and 2, and one private transportation mode, 3, and all individuals share the same choice set.

The first nine observations of the raw data set are shown in Figure 18.1.

Figure 18.1 Initial Choice Data

The following statements transform the data according to MDC procedure requirements:

```
data newdata(keep=pid decision mode ttime);
    set origdata;
    array tvec{3} ttime1 - ttime3;
    retain pid 0;
    pid + 1;
    do i = 1 to 3;
        mode = i;
        ttime = tvec{i};
        decision = ( choice = i );
        output;
    end;
run;
```

The first nine observations of the transformed data set are shown in Figure 18.2.

Figure 18.2 Transformed Modal Choice Data

The decision variable, decision, must have one nonzero value for each decision maker that corresponds to the actual choice. When the RANK option is specified, the decision variable must contain rank data. For more details, see the section "MODEL Statement" on page 1011. The following SAS statements estimate the conditional logit model by using maximum likelihood:

```
proc mdc data=newdata;
    model decision = ttime /
                type=clogit
                nchoice=3
                optmethod=qn
                covest=hess;
    id pid;
run;
```

The MDC procedure enables different individuals to have different choice sets. When all individuals have the same choice set, the NCHOICE= option can be used instead of the CHOICE= option. However, the NCHOICE= option is not allowed when a nested logit model is estimated. When the NCHOICE= option is specified, the choices are generated as 1, ..., number. For more flexible alternatives (for example, 1, 3, 6, 8), you need to use the CHOICE= option. The choice variable must have integer values.

The OPTMETHOD=QN option specifies the quasi-Newton optimization technique. The covariance matrix of the parameter estimates is obtained from the Hessian matrix because COVEST=HESS is specified. You can also specify COVEST=OP or COVEST=QML. See the section "MODEL Statement" on page 1011 for more details.

The MDC procedure produces a summary of model estimation displayed in Figure 18.3. Since there are multiple observations for each individual, the "Number of Cases" (150)—that is, the total number of choices faced by all individuals—is larger than the number of individuals, "Number of Observations" (50).

Figure 18.3 Estimation Summary Table

Figure 18.4 shows the frequency distribution of the three choice alternatives. In this example, mode 2 is most frequently chosen.

Figure 18.4 Choice Frequency

The MDC procedure computes nine goodness-of-fit measures for the discrete choice model. Seven of them are pseudo-R-square measures based on the null hypothesis that all coefficients except for an intercept term are zero (Figure 18.5). McFadden's likelihood ratio index (LRI) is the smallest in value. For more details, see the section "Model Fit and Goodness-of-Fit Statistics" on page 1034.

Figure 18.5 Likelihood Ratio Test and R-Square Measures

Finally, the parameter estimate is displayed in Figure 18.6.

Figure 18.6 Parameter Estimate of Conditional Logit

The predicted choice probabilities are produced using the OUTPUT statement:

```
output out=probdata pred=p;
```

The parameter estimates can be used to forecast the choice probability of individuals that are not in the input data set. To do so, you need to append to the input data set extra observations whose values of the dependent variable decision are missing, since these extra observations are not supposed to be used in the estimation stage. The identification variable id must have values that are not used in the existing observations. The output data set, probdata, contains a new variable, p, in addition to input variables in the data set, setdata.

The following statements forecast the choice probability of individuals that are not in the input data set:

```
data extra;
    input pid mode decision ttime;
datalines;
51   1   .    5.0
51   2   .   15.0
51   3   .   14.0
;

data extdata;
    set newdata extra;
run;

proc mdc data=extdata;
    model decision = ttime /
                type=clogit
                covest=hess
                nchoice=3;
    id pid;
    output out=probdata pred=p;
run;

proc print data=probdata( where=( pid >= 49 ) );
    var mode decision p ttime;
    id pid;
run;
```

The last nine observations from the forecast dataset (probdata) are displayed in Figure 18.7. It is expected that the decision maker will choose mode “1” based on predicted probabilities for all modes.

Figure 18.7 Out-of-Sample Mode Choice Forecast

Nested Logit Modeling

A more general model can be speci ed using the nested logit model.

Consider, for example, the following random utility function:

$$U_{ij} \; D \; x_{ij} \quad C \quad _{ij} \quad j \; D \; 1; : : : ; 3$$

Suppose the set of all alternatives indexed by j is partitioned into K nests, $B_1; \ldots; B_K$. The nested logit model is obtained by assuming that the error term in the utility function has the GEV cumulative distribution function

$$\exp\left(-\sum_{k=1}^{K}\left(\sum_{j \in B_k} \exp\left(-\frac{\varepsilon_j}{\lambda_k}\right)\right)^{\lambda_k}\right)$$

where λ_k is a measure of a degree of independence among the alternatives in nest k. When $\lambda_k = 1$ for all k, the model reduces to the standard logit model.

Since the public transportation modes, 1 and 2, tend to be correlated, these two choices can be grouped together. The decision tree displayed in Figure 18.8 is constructed.

Figure 18.8 Decision Tree for Model Choice

The two-level decision tree is specified in the NEST statement. The NCHOICE= option is not allowed for nested logit estimation. Instead, the CHOICE= option needs to be specified, as in the following statements:

```
/*-- nested logit estimation --          */
proc mdc data=newdata;
    model decision = ttime /
                type=nlogit
                choice=(mode 1 2 3)
                covest=hess;
    id pid;
    utility u(1,) = ttime;
    nest level(1) = (1 2 @ 1, 3 @ 2),
         level(2) = (1 2 @ 1);
run;
```

In Figure 18.9, estimates of the inclusive value parameters, INC_L2G1C1 and INC_L2G1C2, are indicative of a nested model structure. See the section "Nested Logit" on page 1030 and the section "Decision Tree and Nested Logit" on page 1032 for more details about inclusive values.

Figure 18.9 Two-Level Nested Logit Estimates

The nested logit model is estimated with the restriction INC_L2G1C1 = INC_L2G1C2 by specifying the SAMESCALE option, as in the following statements:

```
/*-- nlogit with samescale option --                    */
proc mdc data=newdata;
    model decision = ttime /
                type=nlogit
                choice=(mode 1 2 3)
                samescale
                covest=hess;
    id pid;
    utility u(1,) = ttime;
    nest level(1) = (1 2 @ 1, 3 @ 2),
           level(2) = (1 2 @ 1);
run;
```

The estimation result is displayed in Figure 18.10.

Figure 18.10 Nested Logit Estimates with One Dissimilarity Parameter

The nested logit model is equivalent to the conditional logit model if INC_L2G1C1 = INC_L2G1C2 = 1. You can verify this relationship by estimating a constrained nested logit model as shown in the following statements. (See the section "RESTRICT Statement" on page 1020 for details about imposing linear restrictions on parameter estimates.)

```
/*-- constrained nested logit estimation --              */
proc mdc data=newdata;
    model decision = ttime /
                type=nlogit
                choice=(mode 1 2 3)
                covest=hess;
```

```
    id pid;
    utility u(1,) = ttime;
    nest level(1) = (1 2 @ 1, 3 @ 2),
         level(2) = (1 2 @ 1);
    restrict INC_L2G1C1 = 1, INC_L2G1C2 =1;
run;
```

The parameter estimates and the active linear constraints for the constrained nested logit model are displayed in Figure 18.11.

Figure 18.11 Constrained Nested Logit Estimates

Multivariate Normal Utility Function

Consider the random utility function

$$U_{ij} = \text{ttime}_{ij}\,\gamma + \epsilon_{ij};\quad j = 1,2,3$$

where

$$\begin{bmatrix}\epsilon_{i1}\\ \epsilon_{i2}\\ \epsilon_{i3}\end{bmatrix} \sim N\left(0, \begin{bmatrix}1 & \rho_{21} & 0\\ \rho_{21} & 1 & 0\\ 0 & 0 & 1\end{bmatrix}\right)$$

The correlation coefficient (ρ_{21}) between U_{i1} and U_{i2} represents commonly neglected attributes of public transportation modes, 1 and 2. The following SAS statements estimate this trinomial probit model:

```
/*-- homoscedastic mprobit --        */
proc mdc data=newdata;
    model decision = ttime /
                type=mprobit
                nchoice=3
                unitvariance=(1 2 3)
```

```
                    covest=hess;
    id pid;
run;
```

The UNITVARIANCE=(1 2 3) option specifies that the random component of utility for each of these choices has unit variance. If the UNITVARIANCE= option is specified, it needs to include at least two choices. The results of this constrained multinomial probit model estimation are displayed in Figure 18.12 and Figure 18.13. The test for time = 0 is rejected at the 1% significance level.

Figure 18.12 Constrained Probit Estimation Summary

Figure 18.13 Multinomial Probit Estimates with Unit Variances

HEV and Multinomial Probit: Heteroscedastic Utility Function

When the stochastic components of utility are heteroscedastic and independent, you can model the data by using an HEV or a multinomial probit model. The HEV model assumes that the utility of alternative j for each individual i has heteroscedastic random components,

$$U_{ij} = V_{ij} + \varepsilon_{ij}$$

where the cumulative distribution function of the Gumbel distributed ε_{ij} is

$$F(\varepsilon_{ij}) = \exp(-\exp(-\varepsilon_{ij}/\lambda_j))$$

Note that the variance of ε_{ij} is $\frac{1}{6}\pi^2\lambda_j^2$. Therefore, the error variance is proportional to the square of the scale parameter λ_j. For model identification, at least one of the scale parameters must be normalized to 1. The following SAS statements estimate an HEV model under a unit scale restriction for "1" ($\lambda_1 = 1$):

```
/*-- hev with gauss-laguerre method --*/
proc mdc data=newdata;
    model decision = ttime /
                type=hev
                nchoice=3
                hev=(unitscale=1, integrate=laguerre)
                covest=hess;
    id pid;
run;
```

The results of computation are presented in Figure 18.14 and Figure 18.15.

Figure 18.14 HEV Estimation Summary

Figure 18.15 HEV Parameter Estimates

The parameters SCALE2 and SCALE3 in the output correspond to the estimates of the scale parameters λ_2 and λ_3, respectively.

Note that the estimate of the HEV model is not always stable because computation of the log-likelihood function requires numerical integration. Bhat (1995) proposed the Gauss-Laguerre method. In general, the log-likelihood function value of HEV should be larger than that of conditional logit because HEV models include the conditional logit as a special case. However, in this example the reverse is true (–33.414 for the HEV model, which is less than –33.321 for the conditional logit model). (See Figure 18.14 and Figure 18.3.) This indicates that the Gauss-Laguerre approximation to the true probability is too coarse. You can see how well the Gauss-Laguerre method works by specifying a unit scale restriction for all modes, as in the following statements, since the HEV model with the unit variance for all modes reduces to the conditional logit model:

```
/*-- hev with gauss-laguerre and unit scale --                    */
proc mdc data=newdata;
     model decision = ttime /
                      type=hev
                      nchoice=3
                      hev=(unitscale=1 2 3, integrate=laguerre)
                      covest=hess;
     id pid;
run;
```

Figure 18.16 shows that the ttime coef cient is not close to that of the conditional logit model.

Figure 18.16 HEV Estimates with All Unit Scale Parameters

There is another option of specifying the integration method. The INTEGRATE=HARDY option uses the adaptive Romberg-type integration method. The adaptive integration produces much more accurate probability and log-likelihood function values, but often it is not practical to use this method of analyzing the HEV model because it requires excessive CPU time. The following SAS statements produce the HEV estimates by using the adaptive Romberg-type integration method:

```
/*-- hev with adaptive integration --                             */
proc mdc data=newdata;
     model decision = ttime /
                      type=hev
                      nchoice=3
                      hev=(unitscale=1, integrate=hardy)
                      covest=hess;
     id pid;
run;
```

The results are displayed in Figure 18.17 and Figure 18.18.

Figure 18.17 HEV Estimation Summary Using Alternative Integration Method

Figure 18.18 HEV Estimates Using Alternative Integration Method

With the INTEGRATE=HARDY option, the log-likelihood function value of the HEV model, $-3:026$, is greater than that of the conditional logit model, $-3:321$. (See Figure 18.17 and Figure 18.3.)

When you impose unit scale restrictions on all choices, as in the following statements, the HEV model gives the same estimates as the conditional logit model. (See Figure 18.19 and Figure 18.6.)

```
/*-- hev with adaptive integration and unit scale --                    */
proc mdc data=newdata;
     model decision = ttime /
                      type=hev
                      nchoice=3
                      hev=(unitscale=1 2 3, integrate=hardy)
                      covest=hess;
     id pid;
run;
```

Figure 18.19 Alternative HEV Estimates with Unit Scale Restrictions

For comparison, the following statements estimate a heteroscedastic multinomial probit model by imposing a zero restriction on the correlation parameter, $\rho_{31} = 0$. The MDC procedure requires normalization of at least two of the error variances in the multinomial probit model. Also, for identification, the correlation parameters associated with a unit normalized variance are restricted to be zero. When the UNITVARIANCE= option is specified, the zero restriction on correlation coefficients applies to the last choice of the list. In the following statements, the variances of the first and second choices are normalized. The UNITVARIANCE=(1 2) option imposes additional restrictions that $\omega_{32} = \omega_{21} = 0$. The default for the UNITVARIANCE= option is the last two choices (which would have been equivalent to UNITVARIANCE=(2 3) for this example). The result is presented in Figure 18.20.

The utility function can be defined as

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

where

$$\epsilon_i \sim N\left(0, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \omega_3^2 \end{bmatrix}\right)$$

```
/*-- mprobit estimation --*/
proc mdc data=newdata;
    model decision = ttime /
                type=mprobit
                nchoice=3
                unitvariance=(1 2)
                covest=hess;
    id pid;
    restrict RHO_31 = 0;
run;
```

Figure 18.20 Heteroscedastic Multinomial Probit Estimates

Note that in the output the estimates of standard errors and correlations are denoted by STD_i and RHO_ij, respectively. In this particular case the first two variances (STD_1 and STD_2) are normalized to one, and corresponding correlations (RHO_21 and RHO_32) are set to zero, so they are not listed among parameter estimates.

Parameter Heterogeneity: Mixed Logit

One way of modeling unobserved heterogeneity across individuals in their sensitivity to observed exogenous variables is to use the mixed logit model with a random parameters or random coefficients specification. The probability of choosing alternative j is written as

$$P_i(j) = \frac{\exp(x_{ij}'\beta)}{\sum_{k=1}^{J}\exp(x_{ik}'\beta)}$$

where β is a vector of coefficients that varies across individuals and x_{ij} is a vector of exogenous attributes.

For example, you can specify the distribution of the parameter β to be the normal distribution.

The mixed logit model uses a Monte Carlo simulation method to estimate the probabilities of choice. There are two simulation methods available. If the RANDNUM=PSEUDO option is specified in the MODEL statement, pseudo-random numbers are generated; if the RANDNUM=HALTON option is specified, Halton quasi-random sequences are used. The default value is RANDNUM=HALTON.

You can estimate the model with normally distributed random coef cients of ttime with the following SAS statements:

```
/*-- mixed logit estimation --                    */
proc mdc data=newdata type=mixedlogit;
     model decision = ttime /
                      nchoice=3
                      mixed=(normalparm=ttime);
     id pid;
run;
```

Let $_m$ and $_s$ be mean and scale parameters, respectively, for the random coef cient of ttime. The relevant utility function is

$$U_{ij} \quad D \quad \text{ttime}_{ij} \quad C \quad _{ij}$$

where $D \ _m C \ _s \ (\ _m$ and $_s$ are xed mean and scale parameters, respectively). The stochastic component, , is assumed to be standard normal since the NORMALPARM= option is given. Alternatively, the UNIFORMPARM= or LOGNORMALPARM= option can be speci ed. The LOGNORMALPARM= option is useful when nonnegative parameters are being estimated. The NORMALPARM=, UNIFORMPARM=, and LOGNORMALPARM= variables must be included in the right-hand side of the MODEL statement. See the section "Mixed Logit Model" on page 1026 for more details. To estimate a mixed logit model by using the transportation mode choice data, the MDC procedure requires the MIXED= option for random components. Results of the mixed logit estimation are displayed in Figure 18.21.

Figure 18.21 Mixed Logit Model Parameter Estimates

Note that the parameter ttime_M corresponds to the constant mean parameter $_m$ and the parameter ttime_S corresponds to the constant scale parameter $_s$ of the random coef cient .

Syntax: MDC Procedure

The MDC procedure is controlled by the following statements:

```
PROC MDC options ;
   MDCDATA options ;
   BOUNDS bound1 < , bound2 . . . > ;
   BY variables ;
   CLASS options ;
   ID variable ;
   MODEL dependent variables = regressors / options ;
   NEST LEVEL(value) = ((values)@(value),. . . , (values)@(value)) ;
   NLOPTIONS options ;
   OUTPUT options ;
   RESTRICT restriction1 < , restriction2 . . . > ;
   TEST options ;
   UTILITY  U() = variables, . . . , U() = variables  ;
```

Functional Summary

Table 18.2 summarizes the statements and options used with the MDC procedure.

Table 18.2 MDC Functional Summary

Description	Statement	Option
Data Set Options		
Formats the data for use by PROC MDC	MDCDATA	
Speci es the input data set	MDC	DATA=
Speci es the output data set for CLASS STATEMENT	CLASS	OUT =
Writes parameter estimates to an output data set	MDC	OUTEST=
Includes covariances in the OUTEST= data set	MDC	COVOUT
Writes linear predictors and predicted probabilities to an output data set	OUTPUT	OUT=
Declaring the Role of Variables		
Speci es the ID variable	ID	
Speci es BY-group processing variables	BY	
Printing Control Options		
Requests all printing options	MODEL	ALL
Displays correlation matrix of the estimates	MODEL	CORRB
Displays covariance matrix of the estimates	MODEL	COVB
Displays detailed information about optimization iterations	MODEL	ITPRINT
Suppresses all displayed output	MODEL	NOPRINT

Description	Statement	Option
Model Estimation Options		
Specifies the choice variables	MODEL	CHOICE=()
Specifies the convergence criterion	MODEL	CONVERGE=
Specifies the type of covariance matrix	MODEL	COVEST=
Specifies the starting point of the Halton sequence	MODEL	HALTONSTART=
Specifies options specific to the HEV model	MODEL	HEV=()
Sets the initial values of parameters used by the iterative optimization algorithm	MODEL	INITIAL=()
Specifies the maximum number of iterations	MODEL	MAXITER=
Specifies the options specific to mixed logit	MODEL	MIXED=()
Specifies the number of choices for each person	MODEL	NCHOICE=
Specifies the number of simulations	MODEL	NSIMUL=
Specifies the optimization technique	MODEL	OPTMETHOD=
Specifies the type of random number generators	MODEL	RANDNUM=
Specifies that initial values are generated using random numbers	MODEL	RANDINIT
Specifies the rank dependent variable	MODEL	RANK
Specifies optimization restart options	MODEL	RESTART=()
Specifies a restriction on inclusive parameters	MODEL	SAMESCALE
Specifies a seed for pseudo-random number generation	MODEL	SEED=
Specifies a stated preference data restriction on inclusive parameters	MODEL	SPSCALE
Specifies the type of the model	MODEL	TYPE=
Specifies normalization restrictions on multinomial probit error variances	MODEL	UNITVARIANCE=()
Controlling the Optimization Process		
Specifies upper and lower bounds for the parameter estimates	BOUNDS	
Specifies linear restrictions on the parameter estimates	RESTRICT	
Specifies nonlinear optimization options	NLOPTIONS	
Nested Logit Related Options		
Specifies the tree structure	NEST	LEVEL()=
Specifies the type of utility function	UTILITY	U()=
Output Control Options		
Outputs predicted probabilities	OUTPUT	P=
outputs estimated linear predictor	OUTPUT	XBETA=
Test Request Options		
Requests Wald, Lagrange multiplier, and likelihood ratio tests	TEST	ALL

Description	Statement	Option
Requests the Wald test	TEST	WALD
Requests the Lagrange multiplier test	TEST	LM
Requests the likelihood ratio test	TEST	LR

PROC MDC Statement

PROC MDC options ;

The following options can be used in the PROC MDC statement.

DATA=SAS-data-set
: specifies the input SAS data set. If the DATA= option is not specified, PROC MDC uses the most recently created SAS data set.

OUTEST=SAS-data-set
: names the SAS data set that the parameter estimates are written to. See "OUTEST= Data Set" later in this chapter for information about the contents of this data set.

COVOUT
: writes the covariance matrix for the parameter estimates to the OUTEST= data set. This option is valid only if the OUTEST= option is specified.

In addition, any of the following MODEL statement options can be specified in the PROC MDC statement, which is equivalent to specifying the option for the MODEL statement: ALL, CONVERGE=, CORRB, COVB, COVEST=, HALTONSTART=, ITPRINT, MAXITER=, NOPRINT, NSIMUL=, OPTMETHOD=, RANDINIT, RANK, RESTART=, SAMESCALE, SEED=, SPSCALE, TYPE=, and UNITVARIANCE=.

MDCDATA Statement

MDCDATA options < / OUT= SAS-data-set > ;

The MDCDATA statement prepares data for use by PROC MDC when the choice-specific information is stored in multiple variables (for example, see Figure 18.1 in the section "Conditional Logit: Estimation and Prediction" on page 991).

VARLIST (name1 = (var1 var2 . . .) name2 = (var1 var2 . . .) . . .)
: creates *name* variables from a multiple-variable list of choice alternatives in parentheses. The choice-specific dummy variables are created for the first set of multiple variables. At least one set of multiple variables must be specified. The order of *var*(1 var2 ...) in the VARLIST option determines the numbering of the alternative; that is, *var*1 corresponds to alternative 1, *var*2 corresponds to alternative 2, and so on.

SELECT=(variable)
: specifies a variable that contains choices for each individual. The SELECT= variable needs to be a character-type variable, with values that match variable names in the first VARLIST option: name1=(var1 var2 ...)

ID=(name)
: creates a variable that identifies each individual.

ALT=(name)
: identifies selection alternatives for each individual.

DECVAR=(name)
: creates a 0/1 variable that indicates the choice made for each individual.

OUT=SAS-data-set
: specifies a SAS data set to which modified data are output.

BOUNDS Statement

BOUNDS bound1 < , bound2 . . . > ;

The BOUNDS statement imposes simple boundary constraints on the parameter estimates. BOUNDS statement constraints refer to the parameters estimated by the MDC procedure. You can specify any number of BOUNDS statements.

Each bound is composed of parameters, constants, and inequality operators:

item operator item < operator item < operator item . . . > > ;

Each item is a constant, parameter, or list of parameters. Parameters associated with a regressor variable are referred to by the name of the corresponding regressor variable. Each operator is <, >, <=, or >=.

You can use both the BOUNDS statement and the RESTRICT statement to impose boundary constraints; however, the BOUNDS statement provides a simpler syntax for specifying these kinds of constraints. See also the section "RESTRICT Statement" on page 1020.

Lagrange multipliers are reported for all the active boundary constraints. In the displayed output, the Lagrange multiplier estimates are identified with the names Restrict1, Restrict2, and so on. The probability of the Lagrange multipliers is computed using a beta distribution (LaMotte 1994). Nonactive (nonbinding) bounds have no effect on the estimation results and are not noted in the output.

The following BOUNDS statement constrains the estimates of the coefficients of ttime to be negative and the coefficients of x1 through x10 to be between zero and one. This example illustrates the use of parameter lists to specify boundary constraints.

```
bounds  ttime  <  0,
          0  <  x1-x10  <  1;
```

BY Statement

BY variables ;

A BY statement can be used with PROC MDC to obtain separate analyses on observations in groups de ned by the BY variables.

CLASS Statement

CLASS variables ;

The CLASS statement names the classi cation variables to be used in the analysis. Classi cation variables can be either character or numeric.

ID Statement

ID variable ;

The ID statement must be used with PROC MDC to specify the identi cation variable that controls multiple choice-speci c cases. The MDC procedure requires only one ID statement even with multiple MODEL statements.

MODEL Statement

MODEL dependent = regressors < / options > ;

The MODEL statement speci es the dependent variable and independent regressor variables for the regression model. When the nested logit model is estimated, regressors in the UTILITY statement are used for estimation.

The following options can be used in the MODEL statement after a slash (/).

CHOICE=(variables)

CHOICE=(variable numbers)

speci es the variables that contain possible choices for each individual. Choice variables must have integer values. Multiple choice variables are allowed only for nested logit models and must be speci ed in order from the highest level to the lowest level. For example, CHOICE=(upmode, mode) indicates that the nested logit model has two levels. The choices at the upper level are described by the upmode variable, and the choices at the lower level are described by the mode variable. If all possible alternatives are written with the variable name, the MDC procedure checks all values of the choice variable. CHOICE=(X 1 2 3) implies that the value of X should be 1, 2, or 3. On the other hand, the CHOICE=(X) considers all distinctive nonmissing values of X as elements of the choice set.

CONVERGE=number

speci es the convergence criterion. The CONVERGE= option is the same as the ABSGCONV= option in the NLOPTIONS statement. The ABSGCONV= option in the NLOPTIONS statement overrides the CONVERGE= option. The default value is 1E–5.

HALTONSTART= number
speci es the starting point of the Halton sequence. The speci ed number must be a positive integer. The default is HALTONSTART=11.

HEV=(option-list)
speci es options that are used to estimate the HEV model. The HEV model with a unit scale for the alternative 1 is estimated using the following SAS statement:

```
model y = x1 x2 x3 / hev=(unitscale=1);
```

The following options can be used in the HEV= option. These options are listed within parentheses and separated by commas.

INTORDER=number
speci es the number of summation terms for Gaussian quadrature integration. The default is INTORDER=40. The maximum order is limited to 45. This option applies only to the INTEGRATION=LAGUERRE method.

UNITSCALE=number-list
speci es restrictions on scale parameters of stochastic utility components.

INTEGRATE=LAGUERRE | HARDY
speci es the integration method. The INTEGRATE=HARDY option speci es an adaptive integration method, while the INTEGRATE=LAGUERRE option speci es the Gauss-Laguerre approximation method. The default is INTEGRATE=LAGUERRE.

MIXED=(option-list)
speci es options that are used for mixed logit estimation. The mixed logit model with normally distributed random parameters is speci ed as follows:

```
model y = x1 x2 x3 / mixed=(normalparm=x1);
```

The following options can be used in the MIXED= option. The options are listed within parentheses and separated by commas.

LOGNORMALPARM=variables
speci es the variables whose random coef cients are lognormally distributed. LOGNORMAL-PARM= variablesmust be included on the right-hand side of the MODEL statement.

NORMALEC=variables
speci es the error component variables whose coef cients have a normal distribution $N(0, \sigma^2)$.

NORMALPARM=variables
speci es the variables whose random coef cients are normally distributed. NORMALPARM= variablesmust be included on the right-hand side of the MODEL statement.

UNIFORMEC=variables
speci es the error component variables whose coef cients have a uniform distribution $U(-\sqrt{3}\sigma, \sqrt{3}\sigma)$.

UNIFORMPARM=variables
speci es the variables whose random coef cients are uniformly distributed. UNIFORMPARM= variablesmust be included on the right-hand side of the MODEL statement.

NCHOICE=number

specifies the number of choices for multinomial choice models when all individuals have the same choice set. When individuals have different number of choices, the NCHOICE= option is not allowed, and the CHOICE= option should be used. The NCHOICE= and CHOICE= options must not be used simultaneously, and the NCHOICE= option cannot be used for nested logit models.

NSIMUL=number

specifies the number of simulations when the mixed logit or multinomial probit model is estimated. The default is NSIMUL=100. In general, you need a smaller number of simulations with RANDNUM=HALTON than with RANDNUM=PSEUDO.

RANDNUM=value

specifies the type of the random number generator used for simulation. RANDNUM=HALTON is the default. The following option values are allowed:

PSEUDO specifies pseudo-random number generation.

HALTON specifies Halton sequence generation.

RANDINIT

RANDINIT=number

specifies that initial parameter values be perturbed by uniform pseudo-random numbers for numerical optimization of the objective function. The default is $U(-1, 1)$. When the RANDINIT=*r* option is specified, $U(-r, r)$ pseudo-random numbers are generated. The value *r* should be positive. With a RANDINIT or RANDINIT= option, there are pure random searches for a given number of trials (1,000 for conditional or nested logit, and 500 for other models) to get a maximum (or minimum) value of the objective function. For example, when there is a parameter estimate with an initial value of 1, the RANDINIT option adds a generated random number to the initial value and computes an objective function value by using $1 + u$. This option is helpful in finding the initial value automatically if there is no guidance in setting the initial estimate.

RANK

specifies that the dependent variable contain ranks. The numbers must be positive integers starting from 1. When the dependent variable has value 1, the corresponding alternative is chosen. This option is provided only as a convenience to the user; the extra information contained in the ranks is not currently used for estimation purposes.

RESTART=(option-list)

specifies options that are used for reiteration of the optimization problem. When the ADDRANDOM option is specified, the initial value of reiteration is computed using random grid searches around the initial solution, as follows:

```
model y = x1 x2 / type=clogit
    restart=(addvalue=(.01 .01));
```

The preceding SAS statement reestimates a conditional logit model by adding ADDVALUE= values. If the ADDVALUE= option contains missing values, the RESTART= option uses the corresponding estimate from the initial stage. If no ADDVALUE= value is specified for an estimate, a default value equal to (|estimate| * 1e-3) is added to the corresponding estimate from the initial stage. If both the ADDVALUE= and ADDRANDOM(=) options are specified, ADDVALUE= is ignored.

The following options can be used in the RESTART= option. The options are listed within parentheses.

ADDMAXIT= number
: specifies the maximum number of iterations for the second stage of the estimation. The default is ADDMAXIT=100.

ADDRANDOM | ADDRANDOM=value
: specifies random added values to the estimates from the initial stage. With the ADDRANDOM option, $U(-1, 1)$ random numbers are created and added to the estimates obtained in the initial stage. When the ADDRANDOM=*value* option is specified, $U(-r, r)$ random numbers are generated. The restart initial value is determined based on the given number of random searches (1,000 for conditional or nested logit, and 500 for other models).

ADDVALUE=(value-list)
: specifies values added to the estimates from the initial stage. A missing value in the list is considered as a zero value for the corresponding estimate. When the ADDVALUE= option is not specified, default values equal to (|estimate| * 1e-3) are added.

SAMESCALE
: specifies that the parameters of the inclusive values be the same within a group at each level when the nested logit is estimated.

SEED=number
: specifies an initial seed for pseudo-random number generation. The SEED= value must be less than $2^{31} - 1$. If the SEED= value is negative or zero, the time of day from the computer's clock is used to obtain the initial seed. The default is SEED=0.

SPSCALE
: specifies that the parameters of the inclusive values be the same for any choice with only one nested choice within a group, for each level in a nested logit model. This option is useful in analyzing stated preference data.

TYPE=value
: specifies the type of model to be analyzed. The following model types are supported:

CONDITIONLOGIT | CLOGIT | CL	specifies a conditional logit model.
HEV	specifies a heteroscedastic extreme-value model.
MIXEDLOGIT | MXL	specifies a mixed logit model.
MULTINOMPROBIT | MPROBIT | MP	specifies a multinomial probit model.
NESTEDLOGIT | NLOGIT | NL	specifies a nested logit model.

UNITVARIANCE=(number-list)
: specifies normalization restrictions on error variances of multinomial probit for the choices whose numbers are given in the list. If the UNITVARIANCE= option is specified, it must include at least two choices. Also, for identification, additional zero restrictions are placed on the correlation coefficients for the last choice in the list.

COVEST=value
speci es the type of covariance matrix. The following types are supported:

OP	speci es the covariance from the outer product matrix.
HESSIAN	speci es the covariance from the Hessian matrix.
QML	speci es the covariance from the outer product and Hessian matrices.

When COVEST=OP is speci ed, the outer product matrix is used to compute the covariance matrix of the parameter estimates. The COVEST=HESSIAN option produces the covariance matrix by using the inverse Hessian matrix. The quasi-maximum likelihood estimates are computed with COVEST=QML. The default is COVEST=HESSIAN when the Newton-Raphson method is used. COVEST=OP is the default when the OPTMETHOD=QN option is speci ed.

Printing Options

ALL
requests all printing options.

COVB
displays the estimated covariances of the parameter estimates.

CORRB
displays the estimated correlation matrix of the parameter estimates.

ITPRINT
displays the initial parameter estimates, convergence criteria, and constraints of the optimization. At each iteration, the objective function value, the maximum absolute gradient element, the step size, and the slope of search direction are printed. The objective function is the full negative log-likelihood function for the maximum likelihood method. When the ITPRINT option is speci ed and the NLOPTIONS statement is speci ed, all printing options in the NLOPTIONS statement are ignored.

NOPRINT
suppresses all displayed output.

Estimation Control Options

You can also specify detailed optimization options in the NLOPTIONS statement. The OPTMETHOD= option overrides the TECHNIQUE= option in the NLOPTIONS statement. The NLOPTIONS statement is ignored if the OPTMETHOD= option is speci ed.

INITIAL=(initial-values)
START=(initial-values)
speci es initial values for some or all of the parameter estimates. The values speci ed are assigned to model parameters in the same order in which the parameter estimates are displayed in the MDC procedure output.

When you use the INITIAL= option, the initial values in the INITIAL= option must satisfy the restrictions speci ed for the parameter estimates. If they do not, the initial values you specify are adjusted to satisfy the restrictions.

MAXITER=number
sets the maximum number of iterations allowed. The MAXITER= option overrides the MAXITER= option in the NLOPTIONS statement. The default is MAXITER=100.

OPTMETHOD=value
speci es the optimization technique when the estimation method uses nonlinear optimization. The following techniques are supported:

QN	speci es the quasi-Newton method.
NR	speci es the Newton-Raphson method.
TR	speci es the trust region method.

The OPTMETHOD=NR option is the same as the TECHNIQUE=NEWRAP option in the NLOPTIONS statement. For the conditional and nested logit models, the default is OPTMETHOD=NR. For other models, the default is OPTMETHOD=QN.

NEST Statement

NEST LEVEL (level-number)= (choices@choice, . . .) ;

The NEST statement is used when one choice variable contains all possible alternatives and the TYPE=NLOGIT option is speci ed. The decision tree is constructed based on the NEST statement. When the choice set is speci ed using multiple CHOICE= variables in the MODEL statement, the NEST statement is ignored.

Consider the following eight choices that are nested in a three-level tree structure.

Level 1	Level 2	Level 3	top
1	1	1	1
2	1	1	1
3	1	1	1
4	2	1	1
5	2	1	1
6	2	1	1
7	3	2	1
8	3	2	1

You can use the following NEST statement to specify the tree structure displayed in Figure 18.22:

```
nest level(1) = (1 2 3 @ 1, 4 5 6 @ 2, 7 8 @ 3),
     level(2) = (1 2 @ 1, 3 @ 2),
     level(3) = (1 2 @ 1);
```

Figure 18.22 A Three-Level Tree

Note that the decision tree is constructed based on the sequence of rst-level choice set speci cation. Therefore, specifying another order at Level 1 builds a different tree. The following NEST statement builds the tree displayed in Figure 18.23:

```
nest level(1) = (4 5 6 @ 2, 1 2 3 @ 1, 7 8 @ 3),
     level(2) = (1 2 @ 1, 3 @ 2),
     level(3) = (1 2 @ 1);
```

Figure 18.23 An Alternative Three-Level Tree

However, the NEST statement with a different sequence of choice speci cation at higher levels builds the same tree as displayed in Figure 18.22 if the sequence at the rst level is the same:

```
nest level(1) = (1 2 3 @ 1, 4 5 6 @ 2, 7 8 @ 3),
     level(2) = (3 @ 2, 1 2 @ 1),
     level(3) = (1 2 @ 1);
```

The following speci cations are equivalent:

```
nest level(2) = (3 @ 2, 1 2 @ 1)

nest level(2) = (3 @ 2, 1 @ 1, 2 @ 1)

nest level(2) = (1 @ 1, 2 @ 1, 3 @ 2)
```

Since the MDC procedure contains multiple cases for each individual, it is important to keep the data sequence in the proper order. Consider the four-choice multinomial model with one explanatory variable cost:

pid	choice	y	cost
1	1	1	10
1	2	0	25
1	3	0	20
1	4	0	30
2	1	0	15
2	2	0	22
2	3	1	16
2	4	0	25

The order of data needs to correspond to the value of choice. Therefore, the following data set is equivalent to the preceding data:

pid	choice	y	cost
1	2	0	25
1	3	0	20
1	1	1	10
1	4	0	30
2	3	1	16
2	4	0	25
2	1	0	15
2	2	0	22

The two-level nested model is estimated with a NEST statement, as follows:

```
proc mdc data=one type=nlogit;
    model y = cost / choice=(choice);
    id pid;
    utility(1,) = cost;
    nest level(1) = (1 2 3 @ 1, 4 @ 2),
         level(2) = (1 2 @ 1);
run;
```

The tree is constructed as in Figure 18.24.

Figure 18.24 A Two-Level Tree

Another model is estimated if you specify the decision tree as in Figure 18.25. The different nested tree structure is speci ed in the following SAS statements:

```
proc mdc data=one type=nlogit;
     model y = cost / choice=(choice);
     id pid;
     utility u(1,) = cost;
     nest level(1) = (1 @ 1, 2 3 4 @ 2),
            level(2) = (1 2 @ 1);
run;
```

Figure 18.25 An Alternate Two-Level Tree

NLOPTIONS Statement

NLOPTIONS options ;

PROC MDC uses the nonlinear optimization (NLO) subsystem to perform nonlinear optimization tasks. The NLOPTIONS statement speci es nonlinear optimization options. The NLOPTIONS statement must follow the MODEL statement. For a list of all the options of the NLOPTIONS statement, see Chapter 6, “Nonlinear Optimization Methods.”

OUTPUT Statement

OUTPUT options ;

The OUTPUT statement creates a new SAS data set that contains all the variables in the input data set and, optionally, the estimated linear predictors (XBETA) and predicted probabilities (P). The input data set must be sorted by the choice variables within each ID.

OUT=SAS-data-set
: speci es the name of the output data set.

PRED=variable name

P=variable name
: requests the predicted probabilities by naming the variable that contains the predicted probabilities in the output data set.

XBETA=variable name
: names the variable that contains the linear predictor ($x'\beta$) values. However, the XBETA= option is not supported in the nested logit model.

RESTRICT Statement

RESTRICT restriction1 < , restriction2 . . . > ;

The RESTRICT statement imposes linear restrictions on the parameter estimates. You can specify any number of RESTRICT statements.

Each restriction is written as an expression, followed by an equality operator (=) or an inequality operator (<, >, <=, >=), followed by a second expression:

expression operator expression ;

The operator can be =, <, >, <= , or >=.

Restriction expressions can be composed of parameters; multiplication (*); summation (+), and subtraction ( ) operators; and constants. Parameters named in restriction expressions must be among the parameters estimated by the model. Parameters associated with a regressor variable are referred to by the name of the corresponding regressor variable. The restriction expressions must be a linear function of the parameters.

Lagrange multipliers are reported for all the active linear constraints. In the displayed output, the Lagrange multiplier estimates are identi ed with the names Restrict1, Restrict2, and so on. The probability of the Lagrange multipliers is computed using a beta distribution (LaMotte 1994).

The following are examples of using the RESTRICT statement:

```
proc mdc data=one;
model y = x1-x10 /
          type=clogit
          choice=(mode 1 2 3);
id pid;
restrict x1 *2 <= x2 + x3, ;
run;
```

```
proc mdc data=newdata;
model decision = ttime /
            type=mprobit
            nchoice=3
            unitvariance=(1 2)
            covest=hess;
id pid;
restrict RHO_31 = 0, STD_3<=1;
run;
```

TEST Statement

<`label':> TEST <`string':> equation <,equation. . . > </ options> ;

The TEST statement performs Wald, Lagrange multiplier, and likelihood ratio tests of linear hypotheses about the regression parameters in the preceding MODEL statement. Each equation speci es a linear hypothesis to be tested. All hypotheses in one TEST statement are tested jointly. Variable names in the equations must correspond to regressors in the preceding MODEL statement, and each name represents the coef cient of the corresponding regressor. The keyword INTERCEPT refers to the coef cient of the intercept.

The following options can be speci ed after the slash (/):

ALL
requests Wald, Lagrange multiplier, and likelihood ratio tests.

WALD
requests the Wald test.

LM
requests the Lagrange multiplier test.

LR
requests the likelihood ratio test.

The following statements illustrate the use of the TEST statement:

```
proc mdc;
    model decision = x1 x2 / type=clogit
              choice=(mode 1 2 3);
    id pid;
    test x1 = 0, 0.5 * x1 + 2 * x2 = 0;
run;
```

The test investigates the joint hypothesis that

$$_1 D\ 0$$

and

$$0{:}5\ _1 C\ 2\ _2 D\ 0$$

Only linear equality restrictions and tests are permitted in PROC MDC. Tests expressions can be composed only of algebraic operations that use the addition symbol (+), subtraction symbol (–), and multiplication symbol (*).

The TEST statement accepts labels that are reproduced in the printed output. The TEST statement can be labeled in two ways. A TEST statement can be preceded by a label followed by a colon. Alternatively, the keyword TEST can be followed by a quoted string followed by a colon. If both are present, PROC MDC uses the label that precedes the first colon. If no label is present, PROC MDC automatically labels the tests.

UTILITY Statement

UTILITY U (level < , choices >)= variables ;

The UTILITY statement specifies a utility function that can be used in estimating a nested logit model. The U()= option can have two arguments. The first argument contains level information, and the second argument is related to choice information. The second argument can be omitted for the first level when all the choices at the first level share the same variables and the same parameters. However, for any level above the first, the second argument must be provided. The UTILITY statement specifies a utility function while the NEST statement constructs the decision tree.

Consider a two-level nested logit model that has one explanatory variable at level 1. This model can be specified as follows:

```
proc mdc data=one type=nlogit;
    model y = cost / choice=(choice);
    id pid;
    utility u(1,2 3 4) = cost;
    nest level(1) = (1 @ 1, 2 3 4 @ 2),
           level(2) = (1 2 @ 1);
run;
```

You also can specify the following statement because all the variables at the first level share the same explanatory variable, cost:

```
utility u(1,) = cost;
```

The variable, cost, must be listed in the MODEL statement. When the additional explanatory variable, dummy, is included at level 2, another U()= option needs to be specified. Note that the U()= option must specify choices within any level above the first. Thus, it is specified as U(2, 1 2) in the following statements:

```
proc mdc data=one type=nlogit;
    model y = cost dummy / choice=(choice);
    id pid;
    utility u(1,) = cost,
                u(2,1 2) = dummy;
    nest level(1) = (1 @ 1, 2 3 4 @ 2),
           level(2) = (1 2 @ 1);
run;
```

Details: MDC Procedure

Multinomial Discrete Choice Modeling

When the dependent variable takes multiple discrete values, you can use multinomial discrete choice modeling to analyze the data. This section considers models for unordered multinomial data.

Let the random utility function be de ned by

$$U_{ij} = V_{ij} + \varepsilon_{ij}$$

where the subscript i is an index for the individual, the subscript j is an index for the alternative, V_{ij} is a nonstochastic utility function, and ε_{ij} is a random component (error) that captures unobserved characteristics of alternatives or individuals or both. In multinomial discrete choice models, the utility function is assumed to be linear, so that $V_{ij} = x_{ij}'\beta$.

In the conditional logit model, each ε_{ij} for all $j \in C_i$ is distributed independently and identically (iid) with the Type I extreme-value distribution, $\exp(-\exp(-\varepsilon_{ij}))$, also known as the Gumbel distribution.

The iid assumption on the random components of the utilities of the different alternatives can be relaxed to overcome the well-known and restrictive independence from irrelevant alternatives (IIA) property of the conditional logit model. This allows for more flexible substitution patterns among alternatives than the one imposed by the conditional logit model. See the section "Independence from Irrelevant Alternatives (IIA)" on page 1025.

The nested logit model is derived by allowing the random components to be identical but nonindependent. Instead of independent Type I extreme-value errors, the errors are assumed to have a generalized extreme-value distribution. This model generalizes the conditional logit model to allow for particular patterns of correlation in unobserved utility (McFadden 1978).

Another generalization of the conditional logit model, the heteroscedastic extreme-value (HEV) model, is obtained by allowing independent but nonidentical errors distributed with a Type I extreme-value distribution (Bhat 1995). It permits different variances on the random components of utility across the alternatives.

Mixed logit models are also generalizations of the conditional logit model that can represent very general patterns of substitution among alternatives. See the "Mixed Logit Model" on page 1026 section for details.

The multinomial probit (MNP) model is derived when the errors, $(\varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{iJ})$, have a multivariate normal (MVN) distribution. Thus, this model accommodates a very general error structure.

The multinomial probit model requires burdensome computation compared to a family of multinomial choice models derived from the Gumbel distributed utility function, since it involves multi-dimensional integration (with dimension $J - 1$) in the estimation process. In addition, the multinomial probit model requires more parameters than other multinomial choice models. As a result, conditional and nested logit models are used more frequently, even though they are derived from a utility function whose random component is more restrictively de ned than the multinomial probit model.

The event of a choice being made, $y_i = j$, can be expressed using a random utility function

$$U_{ij} = \max_{k \in C_i; k \neq j} U_{ik}$$

where C_i is the choice set of individual i. Individual i chooses alternative j if and only if it provides a level of utility that is greater than or equal to that of any other alternative in his choice set. Then, the probability that individual i chooses alternative j (from among the n_i choices in his choice set C_i) is

$$P_i(j) = P_{ij} = P(\varepsilon_{ij} - \varepsilon_{ik} \geq \max_{k \in C_i}(x_{ik}'\beta - x_{ij}'\beta))$$

Multinomial Logit and Conditional Logit

When explanatory variables contain only individual characteristics, the multinomial logit model is defined as

$$P(y_i = j) = P_{ij} = \frac{\exp(x_i'\beta_j)}{\sum_{k=0}^{J}\exp(x_i'\beta_k)} \quad \text{for } j = 0, \ldots, J$$

where y_i is a random variable that indicates the choice made, x_i is a vector of characteristics specific to the ith individual, and β_j is a vector of coefficients specific to the jth alternative. Thus, this model involves choice-specific coefficients and only individual specific regressors. For model identification, it is often assumed that $\beta_0 = 0$. The multinomial logit model reduces to the binary logit model if $J = 1$.

The ratio of the choice probabilities for alternatives j and l (the odds ratio of alternatives j and l) is

$$\frac{P_{ij}}{P_{il}} = \frac{\exp(x_i'\beta_j)/\sum_{k=0}^{J}\exp(x_i'\beta_k)}{\exp(x_i'\beta_l)/\sum_{k=0}^{J}\exp(x_i'\beta_k)} = \exp(x_i'(\beta_j - \beta_l))$$

Note that the odds ratio of alternatives j and l does not depend on any alternatives other than j and l. For more information, see the section "Independence from Irrelevant Alternatives (IIA)" on page 1025.

The log-likelihood function of the multinomial logit model is

$$L = \sum_{i=1}^{N}\sum_{j=0}^{J} d_{ij} \ln P(y_i = j)$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

This type of multinomial choice modeling has a couple of weaknesses: it has too many parameters (the number of individual characteristics times J), and it is difficult to interpret. The multinomial logit model can be used to predict the choice probabilities, among a given set of $J+1$ alternatives, of an individual with known vector of characteristics x_i.

The parameters of the multinomial logit model can be estimated with the TYPE=CLOGIT option in the MODEL statement; however, this requires modification of the conditional logit model to allow individual specific effects.

The conditional logit model, sometimes called the multinomial logit model, is similarly defined when choice-specific data are available. Using properties of Type I extreme-value (Gumbel) distribution, the probability that individual i chooses alternative j from among the choices in his choice set C_i is

$$P(y_i = j) = P_{ij} = P(\varepsilon_{ij} - \varepsilon_{ik} \geq \max_{k \in C_i; k \neq j}(x_{ik}'\beta - x_{ij}'\beta)) = \frac{\exp(x_{ij}'\beta)}{\sum_{k \in C_i}\exp(x_{ik}'\beta)}$$

where x_{ij} is a vector of attributes specific to the jth alternative as perceived by the ith individual. It is assumed that there are n_i choices in each individual's choice set, C_i.

The log-likelihood function of the conditional logit model is

$$L = \sum_{i=1}^{N} \sum_{j \in C_i} d_{ij} \ln P(y_i = j)$$

The conditional logit model can be used to predict the probability that an individual will choose a previously unavailable alternative, given knowledge of β and the vector x_{ij} of choice-specific characteristics.

Independence from Irrelevant Alternatives (IIA)

The problematic aspect of the conditional logit (and the multinomial logit) model lies in the property of independence from irrelevant alternatives (IIA). The IIA property can be derived from the probability ratio of any two choices. For the conditional logit model,

$$\frac{P_{ij}}{P_{il}} = \frac{\exp(x_{ij}'\beta)}{\exp(x_{il}'\beta)} \Big/ \frac{\sum_{k \in C_i} \exp(x_{ik}'\beta)}{\sum_{k \in C_i} \exp(x_{ik}'\beta)} = \exp(\beta'(x_{ij} - x_{il}))$$

It is evident that the ratio of the probabilities for alternatives j and l does not depend on any alternatives other than j and l. This was also shown to be the case for the multinomial logit model. Thus, for the conditional and multinomial logit models, the ratio of probabilities of any two alternatives is necessarily the same regardless of what other alternatives are in the choice set or what the characteristics of the other alternatives are. This is referred to as the IIA property.

The IIA property is useful from the point of view of estimation and forecasting. For example, it allows the prediction of demand for currently unavailable alternatives. If the IIA property is appropriate for the choice situation being considered, then estimation can be based on the set of currently available alternatives, and then the estimated model can be used to calculate the probability that an individual would choose a new alternative not considered in the estimation procedure. However, the IIA property is restrictive from the point of view of choice behavior. Models that display the IIA property predict that a change in the attributes of one alternative changes the probabilities of the other alternatives proportionately such that the ratios of probabilities remain constant. Thus, cross elasticities due to a change in the attributes of an alternative j are equal for all alternatives $k \neq j$. This particular substitution pattern might be too restrictive in some choice settings.

The IIA property of the conditional logit model follows from the assumption that the random components of utility are identically and independently distributed. The other models in PROC MDC (namely, nested logit, HEV, mixed logit, and multinomial probit) relax the IIA property in different ways.

For an example of Hausman's specification test of IIA assumption, see “Example 18.6: Hausman's Specification Test” on page 1058.

Heteroscedastic Extreme-Value Model

The heteroscedastic extreme-value (HEV) model (Bhat 1995) allows the random components of the utility function to be nonidentical. Specifically, the HEV model assumes independent but nonidentical error terms distributed with the Type I extreme-value distribution. The HEV model allows the variances of the random

components of utility to differ across alternatives. Bhat (1995) argues that the HEV model does not have the IIA property. The HEV model contains the conditional logit model as a special case. The probability that an individual i will choose alternative j from the set C_i of available alternatives is

$$P_i(j) = \int_{-\infty}^{\infty} \prod_{k \in C_i; k \neq j} \exp\left[-\exp\left(\frac{x_{ij}^0 - x_{ik}^0 + \theta_j w}{\theta_k} \right) \right] \exp(-w)\, dw$$

where the choice set C_i has n_i elements and

$$\exp(x) = \exp(-\exp(-x))$$

$$\exp(x) = \exp(-x)\exp(x)$$

are the cumulative distribution function and probability density function of the Type I extreme-value distribution. The variance of the error term for the jth alternative is $\frac{1}{6}\pi^2\theta_j^2$. If the scale parameters, θ_j, of the random components of utility of all alternatives are equal, then this choice probability is the same as that of the conditional logit model. The log-likelihood function of the HEV model can be written as

$$L = \sum_{i=1}^{N} \sum_{j \in C_i} d_{ij} \ln(P_i(j))$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

Since the log-likelihood function contains an improper integral function, it is computationally difficult to get a stable estimate. With the transformation $u = \exp(-w)$, the probability can be written

$$P_i(j) = \int_0^{\infty} \cdots \prod_{k \in C_i; k \neq j} \exp\left[-\exp\left(\frac{x_{ij}^0 - x_{ik}^0 - \theta_j \ln(u)}{\theta_k} \right) \right] \exp(-u)\, du$$

$$= \int_0^{\infty} G_{ij}(u) \exp(-u)\, du$$

Using the Gauss-Laguerre weight function $W(x) = \exp(-x)$, the integration of the log-likelihood function can be replaced with a summation as follows:

$$\int_0^{\infty} G_{ij}(u) \exp(-u)\, du = \sum_{k=1}^{K} w_k G_{ij}(x_k)$$

Weights (w_k) and abscissas (x_k) are tabulated by Abramowitz and Stegun (1970).

Mixed Logit Model

In mixed logit models, an individual's utility from any alternative can be decomposed into a deterministic component $x_{ij}^0\beta$, which is a linear combination of observed variables, and a stochastic component, $\theta_j\varepsilon_{ij}$,

$$U_{ij} = x_{ij}^0\beta + \theta_j\varepsilon_{ij}$$

where x_{ij} is a vector of observed variables that relate to individual i and alternative j, β is a vector of parameters, ε_{ij} is an error component that can be correlated among alternatives and heteroscedastic for each individual, and η_{ij} is a random term with zero mean that is independently and identically distributed over alternatives and individuals. The conditional logit model is derived if you assume ε_{ij} has an iid Gumbel distribution and $V(\varepsilon_{ij}) = 0$.

The mixed logit model assumes a general distribution for ε_{ij} and an iid Gumbel distribution for η_{ij}. Denote the density function of the error component as $f(\varepsilon_{ij} \mid \theta)$, where θ is a parameter vector of the distribution of ε_{ij}. The choice probability of alternative j for individual i is written as

$$P_i(j) = \int Q_i(j \mid \varepsilon_{ij}) f(\varepsilon_{ij} \mid \theta) d\varepsilon_{ij}$$

where the conditional choice probability for a given value of ε_{ij} is the logit

$$Q_i(j \mid \varepsilon_{ij}) = \frac{\exp(x_{ij}'\beta + \varepsilon_{ij})}{\sum_{k \in C_i} \exp(x_{ik}'\beta + \varepsilon_{ik})}$$

Since ε_{ij} is not given, the unconditional choice probability, $P_i(j)$, is the integral of the conditional choice probability, $Q_i(j \mid \varepsilon_{ij})$, over the distribution of ε_{ij}. This model is called "mixed logit" since the choice probability is a mixture of logits with $f(\varepsilon_{ij} \mid \theta)$ as the mixing distribution.

In general, the mixed logit model does not have an exact likelihood function because the probability $P_i(j)$ does not always have a closed form solution. Therefore, a simulation method is used for computing the approximate probability,

$$\hat{P}_i(j) = 1/S \sum_{s=1}^{S} Q_i(j \mid \varepsilon_{ij}^s)$$

where S is the number of simulation replications and $\hat{P}_i(j)$ is a simulated probability. The simulated log-likelihood function is computed as

$$LL = \sum_{i=1}^{N} \sum_{j=1}^{n_i} d_{ij} \ln(\hat{P}_i(j))$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

For simulation purposes, assume that the error component has a specific structure,

$$\varepsilon_{ij} = z_{ij}'\alpha + w_{ij}'\eta$$

where z_{ij} is a vector of observed data and η is a random vector with zero mean and density function $f(\eta)$. The observed data vector z_{ij} of the error component can contain some or all elements of x_{ij}. The component $z_{ij}'\alpha$ induces heteroscedasticity and correlation across unobserved utility components of the alternatives. This allows flexible substitution patterns among the alternatives. The kth element of vector α is distributed as

$$\alpha_k \sim N(0, \sigma_k^2)$$

Therefore, $_k$ can be specified as

$$_k D \quad _k$$

where

$$N.0; 1/$$

or

$$U. \quad \frac{p}{3}; \frac{p}{3}/$$

In addition, is a vector of random parameters (random coefficients). Random coefficients allow heterogeneity across individuals in their sensitivity to observed exogenous variables. The observed data vector, x_{ij}, is a subset of x_{ij} . The following three types of distributions for the random coefficients are supported, where the mth element of is denoted as $_m$:

- Normally distributed coefficient with the mean b_m and spread s_m being estimated.

  $$_m D b_m C s_m \quad \text{and} \quad N.0; 1/$$

- Uniformly distributed coefficient with the mean b_m and spread s_m being estimated. A uniform distribution with mean b and spread s is $U.b \quad s; b C s/$.

  $$_m D b_m C s_m \quad \text{and} \quad U. \quad 1; 1/$$

- Lognormally distributed coefficient. The coefficient is calculated as

  $$_m D \exp.b_m C s_m \quad / \quad \text{and} \quad N.0; 1/$$

  where b_m and s_m are parameters that are estimated.

The estimate of spread for normally, uniformly, and lognormally distributed coefficients can be negative. The absolute value of the estimated spread can be interpreted as an estimate of standard deviation for normally distributed coefficients.

A detailed description of mixed logit models can be found, for example, in Brownstone and Train (1999).

Multinomial Probit

The multinomial probit model allows the random components of the utility of the different alternatives to be nonindependent and nonidentical. Thus, it does not have the IIA property. The increase in the flexibility of the error structure comes at the expense of introducing several additional parameters in the covariance matrix of the errors.

Consider the random utility function

$$U_{ij} \quad D \quad x_{ij}^{0} \quad C \quad _{ij}$$

where the joint distribution of $\varepsilon_{i1}, \varepsilon_{i2}, \ldots, \varepsilon_{iJ}$ is multivariate normal:

$$\begin{bmatrix} \varepsilon_{i1} \\ \varepsilon_{i2} \\ \vdots \\ \varepsilon_{iJ} \end{bmatrix} \sim N(0, \Sigma)$$

$$\Sigma = \sigma_{jk}, \quad j,k = 1,\ldots,J$$

The dimension of the error covariance matrix is determined by the number of alternatives. Given $(x_{i1}, x_{i2}, \ldots, x_{iJ})$, the jth alternative is chosen if and only if $U_{ij} > U_{ik}$ for all $k \neq j$. Thus, the probability that the jth alternative is chosen is

$$P(y_i = j) = P_{ij} = P(\varepsilon_{i1} - \varepsilon_{ij} < x_{ij} - x_{i1}, \ldots, \varepsilon_{iJ} - \varepsilon_{ij} < x_{ij} - x_{iJ})$$

where y_i is a random variable that indicates the choice made. This is a cumulative probability from a $(J-1)$-variate normal distribution. Since evaluation of this probability involves multidimensional integration, it is practical to use a simulation method to estimate the model. Many studies have shown that the simulators proposed by the following authors (henceforth referred to as GHK) perform well: Geweke (1989); Hajivassiliou (1993); Keane (1994). For example, Hajivassiliou, McFadden, and Ruud (1996) compare 13 simulators using 11 different simulation methods and conclude that the GHK simulation method is the most reliable. To compute the probability of the multivariate normal distribution, the recursive simulation method is used. Refer to Hajivassiliou (1993) for more details about GHK simulators.

The log-likelihood function for the multinomial probit model can be written as

$$L = \sum_{i=1}^{N} \sum_{j=1}^{J} d_{ij} \ln P(y_i = j)$$

where

$$d_{ij} = \begin{cases} 1 & \text{if individual } i \text{ chooses alternative } j \\ 0 & \text{otherwise} \end{cases}$$

For identification of the multinomial probit model, two of the diagonal elements of Σ are normalized to 1, and it is assumed that for one of the choices whose error variance is normalized to 1, (say, k), it is also true that $\sigma_{jk} = \sigma_{kj} = 0$ for $j = 1, \ldots, J$ and $j \neq k$. Thus, a model with J alternatives has at most $J(J-1)/2 - 1$ covariance parameters after normalization.

Let D and R be defined as

$$D = \begin{bmatrix} \sigma_1 & 0 & \cdots & 0 \\ 0 & \sigma_2 & \cdots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & & \sigma_J \end{bmatrix}$$

$$R = \begin{bmatrix} 1 & \rho_{12} & \cdots & \rho_{1J} \\ \rho_{21} & 1 & \cdots & \rho_{2J} \\ \vdots & \vdots & \vdots & \vdots \\ \rho_{J1} & \rho_{J2} & & 1 \end{bmatrix}$$

where $\sigma_j^2 = \omega_{jj}$ and $\rho_{jk} = \frac{\omega_{jk}}{\sigma_j \sigma_k}$. Then, for identification, $\sigma_{J-1} = \sigma_J = 1$ and $\rho_{kJ} = \rho_{Jk} = 0$, for all $k \neq J$ can be imposed, and the error covariance matrix is $\Omega = DRD$.

In the standard MDC output, the parameter estimates STD_j and RHO_jk correspond to σ_j and ρ_{jk}.

In principle, the multinomial probit model is fully identified with the preceding normalizations. However, in practice, convergence in applications of the model with more than three alternatives often requires additional restrictions on the elements of Ω.

It must also be noted that the unrestricted structure of the error covariance matrix makes it impossible to forecast demand for a new alternative without knowledge of the new $(J+1)$ by $(J+1)$ error covariance matrix.

Nested Logit

The nested logit model (McFadden 1978, 1981) allows partial relaxation of the assumption of independence of the stochastic components of utility of alternatives. In some choice situations, the IIA property holds for some pairs of alternatives but not all. In these situations, the nested logit model can be used if the set of alternatives faced by an individual can be partitioned into subsets such that the IIA property holds within subsets but not across subsets.

In the nested logit model, the joint distribution of the errors is generalized extreme value (GEV). This is a generalization of the Type I extreme-value distribution that gives rise to the conditional logit model. Note that all ε_{ij} within each subset are correlated with each other. Refer to McFadden (1978, 1981) for details.

Nested logit models can be described analytically following the notation of McFadden (1981). Assume that there are L levels, with 1 representing the lowest and L representing the highest level of the tree. The index of a node at level h in the tree is a pair (j_h, η_h), where $\eta_h = (j_{h+1}, \ldots, j_L)$ is the index of the adjacent node at level $h+1$. Thus, the primitive alternatives, at level 1 in the tree, are indexed by vectors $(j_2, \ldots, j_L)$, and the alternative nodes at level L are indexed by integers j_L. The choice set C_h is the set of primitive alternatives (at level 1) that belong to branches below the node η_h. The notation C_h is also used to denote a set of indices j_h such that (j_h, η_h) is a node immediately below η_h. Note that C_0 is a set with a single element, while C_L represents a choice set that contains all possible alternatives. As an example, consider the circled node at level 1 in Figure 18.26. Since it stems from node $\eta_2 = 11$, and since it is the second node stemming from 11, $j_h = 2$, its index is $\eta_1 = \eta_0 = (j_h, \eta_h) = 211$. Similarly, $C_{11} = \{111, 211, 311\}$ contains all the possible choices below 11.

Although this notation is useful for writing closed-form solutions for probabilities, the MDC procedure allows a more flexible definition of indices. See the section "NEST Statement" on page 1016 for more details about how to describe decision trees within the MDC procedure.

Figure 18.26 Node Indices for a Three-Level Tree

Let $x_{i|j_h,h}^{(h)}$ denote the vector of observed variables for individual i common to the alternatives below node j_h at level h. The probability of choice at level h has a closed-form solution and is written

$$P_i(j_h|j_{h+1}) = \frac{\exp\left[x_{i|j_h,h}^{(h)\prime}\beta^{(h)} + \theta_h \sum_{k \in C_{i|j_h,h}} I_{k;j_h,h}\right]}{\sum_{j \in C_{i|j_{h+1},h+1}} \exp\left[x_{i|j,h}^{(h)\prime}\beta^{(h)} + \theta_h \sum_{k \in C_{i|j,h}} I_{k;j,h}\right]}; \quad h = 2, \ldots, L$$

where $I_{j,h}$ is the inclusive value (at level $h-1$) of the branch below node j_h and is defined recursively as follows:

$$I_{j,h} = \ln\left\{\sum_{j \in C_{i|j_{h+1},h+1}} \exp\left[x_{i|j,h}^{(h)\prime}\beta^{(h)} + \theta_h \sum_{k \in C_{i|j,h}} I_{k;j,h}\right]\right\}$$

$$\theta_1 \quad \theta_{k;1} \quad \theta_{k;L-1}$$

The inclusive value $I_{j,h}$ denotes the average utility that the individual can expect from the branch below j_h. The dissimilarity parameters or inclusive value parameters ($\theta_{k;j,h}$) are the coefficients of the inclusive values and have values between 0 and 1 if nested logit is the correct model specification. When they all take value 1, the nested logit model is equivalent to the conditional logit model.

At decision level 1, there is no inclusive value; that is, $\theta_0 = 0$. Therefore, the conditional probability is

$$P_i(j_1|j_2) = \frac{\exp\left[x_{i|j_1,1}^{(1)\prime}\beta^{(1)}\right]}{\sum_{j \in C_{i|j_2,1}} \exp\left[x_{i|j,1}^{(1)\prime}\beta^{(1)}\right]}$$

The log-likelihood function at level h can then be written

$$L^{(h)} = \sum_{i=1}^{N} \sum_{j_{h+1} \in C_{i;j_{h+2}}} \sum_{j \in C_{i;j_{h+1}}} y_{i;j_{h+1}} \ln P(C_{i;j_{h+1}}|C_{i;j_{h+2}})$$

where $y_{i;j\ h0}$ is an indicator variable that has the value of 1 for the selected choice. The full log-likelihood function of the nested logit model is obtained by adding the conditional log-likelihood functions at each level:

$$L\ D\ \sum_{hD\,1}^{X} L^{.h/}$$

Note that the log-likelihood functions are computed from conditional probabilities when . The nested logit model is estimated using the full information maximum likelihood method.

Decision Tree and Nested Logit

You can view choices as a decision tree and model the decision tree by using the nested logit model. You need to use either the NEST statement or the CHOICE= option of the MODEL statement to specify the nested tree structure. Additionally, you need to identify which explanatory variables are used at each level of the decision tree. These explanatory variables are arguments for what is called a utility function. The utility function is speci ed using UTILITY statements. For example, consider a two-level decision tree. The tree structure is displayed in Figure 18.27.

Figure 18.27 Two-Level Decision Tree

A nested logit model with two levels can be speci ed using the following SAS statements:

```
proc mdc data=one type=nlogit;
    model decision = x1 x2 x3 x4 x5 /
          choice=(upmode 1 2, mode 1 2 3 4 5);
    id pid;
    utility u(1, 3 4 5 @ 2) = x1 x2,
            u(1, 1 2 @ 1) = x3 x4,
            u(2, 1 2) = x5;
run;
```

The DATA=one data set should be arranged as follows:

obs	pid	upmode	mode	x1	x2	x3	x4	x5	decision
1	1	1	1	#	#	#	#	#	1
2	1	1	2	#	#	#	#	#	0
3	1	2	3	#	#	#	#	#	0
4	1	2	4	#	#	#	#	#	0
5	1	2	5	#	#	#	#	#	0
6	2	1	1	#	#	#	#	#	0
7	2	1	2	#	#	#	#	#	0
8	2	2	3	#	#	#	#	#	0
9	2	2	4	#	#	#	#	#	0
10	2	2	5	#	#	#	#	#	1

All model variables, x1 through x5, are speci ed in the UTILITY statement. It is required that entries denoted as # have values for model estimation and prediction. The values of the level 2 utility variable x3 should be the same for all the primitive (level 1) alternatives below node 1 at level 2 and, similarly, for all the primitive alternatives below node 2 at level 2. In other words, x5 should have the same value for primitive alternatives 1 and 2 and, similarly, it should have the same value for primitive alternatives 3, 4, and 5. More generally, the values of any level 2 or higher utility function variables should be constant across primitive alternatives under each node for which the utility function applies. Since PROC MDC expects this to be the case, it uses the values of x5 only for the primitive alternatives 1 and 3, ignoring the values for the primitive alternatives 2, 4, and 5. Thus, PROC MDC uses the values of the utility function variable only for the primitive alternatives that come rst under each node for which the utility function applies. This behavior applies to any utility function variables that are speci ed above the rst level. The choice variable for level 2 (upmode) should be placed before the rst-level choice variable (mode) when the CHOICE= option is speci ed. Alternatively, the NEST statement can be used to specify the decision tree. The following SAS statements t the same nested logit model:

```
proc mdc data=a type=nlogit;
    model decision = x1 x2 x3 x4 x5 /
                choice=(mode 1 2 3 4 5);
    id pid;
    utility u(1, 3 4 5 @ 2) = x1 x2,
            u(1, 1 2 @ 1) = x3 x4,
            u(2, 1 2) = x5;
    nest level(1) = (1 2 @ 1, 3 4 5 @ 2),
         level(2) = (1 2 @ 1);
run;
```

The U(1, 3 4 5 @ 2)= option speci es three choices, 3, 4, and 5, at level 1 of the decision tree. They are connected to the upper branch 2. The speci ed variables (x1 and x2) are used to model this utility function. The bottom level of the decision tree is level 1. All variables in the UTILITY statement must be included in the MODEL statement. When all choices at the rst level share the same variables, you can omit the second argument of the U()= option for that level. However, U(1,) = x1 x2 is not equivalent to the following statements:

```
u(1, 3 4 5 @ 2) = x1 x2;
u(1, 1 2 @ 1) = x1 x2;
```

The CHOICE= variables need to be speci ed from the top to the bottom level. To forecast demand for new products, stated preference data are widely used. Stated preference data are attractive for market researchers

because attribute variations can be controlled. Hensher (1993) explores the advantage of combining revealed preference (market data) and stated preference data. The scale factor μ can be estimated using the nested logit model with the decision tree structure displayed in Figure 18.28.

Figure 18.28 Decision Tree for Revealed and Stated Preference Data

Example SAS statements are as follows:

```
proc mdc data=a type=nlogit;
    model decision = x1 x2 x3 /
                spscale
                choice=(mode 1 2 3 4 5 6);
    id pid;
    utility u(1,) = x1 x2 x3;
    nest level(1) = (1 2 3 @ 1, 4 @ 2, 5 @ 3, 6 @ 4),
           level(2) = (1 2 3 4 @ 1);
run;
```

The SPSCALE option specifies that parameters of inclusive values for nodes 2, 3, and 4 at level 2 be the same. When you specify the SAMESCALE option, the MDC procedure imposes the same coefficient of inclusive values for choices 1–4.

Model Fit and Goodness-of-Fit Statistics

McFadden (1974) suggests a likelihood ratio index that is analogous to the R-square in the linear regression model:

$$R^2_M = 1 - \frac{\ln L}{\ln L_0}$$

where L is the maximum of the log-likelihood function and L_0 is the maximum of the log-likelihood function when all coefficients, except for an intercept term, are zero. McFadden's likelihood ratio index is bounded by 0 and 1.

Estrella (1998) proposes the following requirements for a goodness-of-fit measure to be desirable in discrete choice modeling:

- The measure must take values in $[0, 1]$, where 0 represents no fit and 1 corresponds to perfect fit.

- The measure should be directly related to the valid test statistic for the signi cance of all slope coef cients.
- The derivative of the measure with respect to the test statistic should comply with corresponding derivatives in a linear regression.

Estrella's measure is written as

$$R^2_{E1} = 1 - \left(\frac{\ln L}{\ln L_0}\right)^{-2/N \ln L_0}$$

Estrella suggests an alternative measure,

$$R^2_{E2} = 1 - \left(\frac{\ln L}{\ln L_0}\right)^{-2/N \ln L_0}$$

where $\ln L_0$ is computed with null parameter values, N is the number of observations used, and k represents the number of estimated parameters.

Other goodness-of- t measures are summarized as follows:

$$R^2_{CU1} = 1 - \left(\frac{L_0}{L}\right)^{\frac{2}{N}} \quad \text{.Cragg-Uhler 1}$$

$$R^2_{CU2} = \frac{1 - (L_0/L)^{\frac{2}{N}}}{1 - L_0^{\frac{2}{N}}} \quad \text{.Cragg-Uhler 2}$$

$$R^2_A = \frac{2\ln L - \ln L_0}{2\ln L - \ln L_0 + N} \quad \text{.Aldrich-Nelson}$$

$$R^2_{VZ} = R^2_A \frac{2\ln L_0 - N}{2\ln L_0} \quad \text{.Veall-Zimmermann}$$

The AIC and SBC are computed as follows:

$$AIC = -2\ln(L) + 2k$$

$$SBC = -2\ln(L) + \ln(n)k$$

where $\ln(L)$ is the log-likelihood value for the model, k is the number of parameters estimated, and n is the number of observations (that is, the number of respondents).

Tests on Parameters

In general, the hypothesis to be tested can be written as

$$H_0: h(\theta) = 0$$

where $h(\theta)$ is an r-by-1 vector-valued function of the parameters given by the r expressions speci ed in the TEST statement.

Let V be the estimate of the covariance matrix of θ. Let $\hat{\theta}$ be the unconstrained estimate of θ, and $\tilde{\theta}$ be the constrained estimate of θ such that $h(\tilde{\theta}) = 0$. Let

$$A(\theta) = \partial h(\theta)/\partial \theta_j$$

Using this notation, the test statistics for the three kinds of tests are computed as follows:

- The Wald test statistic is de ned as

$$W = h(\hat{\theta})' \left\{ \nabla h(\hat{\theta}) \left[\nabla^2 L(\hat{\theta}) \right]^{-1} \nabla h(\hat{\theta})' \right\}^{-1} h(\hat{\theta})$$

  The Wald test is not invariant to reparameterization of the model (Gregory and Veall 1985; Gallant 1987, p. 219). For more information about the theoretical properties of the Wald test, see Phillips and Park (1988).

- The Lagrange multiplier test statistic is

$$LM = \lambda' \nabla^2 L(\tilde{\theta}) \left[\nabla g(\tilde{\theta})' \right]^{-1} \lambda$$

  where λ is the vector of Lagrange multipliers from the computation of the restricted estimate $\tilde{\theta}$

- The likelihood ratio test statistic is

$$LR = 2 \left[L(\hat{\theta}) - L(\tilde{\theta}) \right]$$

  where $\tilde{\theta}$ represents the constrained estimate and L is the concentrated log-likelihood value.

For each kind of test, under the null hypothesis the test statistic is asymptotically distributed as a χ^2 random variable with r degrees of freedom, where r is the number of expressions in the TEST statement. The p-values reported for the tests are computed from the χ^2_r distribution and are only asymptotically valid.

Monte Carlo simulations suggest that the asymptotic distribution of the Wald test is a poorer approximation to its small sample distribution than that of the other two tests. However, the Wald test has the lowest computational cost, since it does not require computation of the constrained estimate $\tilde{\theta}$

The following statements are an example of using the TEST statement to perform a likelihood ratio test:

```
proc mdc;
    model decision = x1 x2 / type=clogit
          choice=(mode 1 2 3);
    id pid;
    test 0.5 * x1 + 2 * x2 = 0 / lr;
run;
```

OUTEST= Data Set

The OUTEST= data set contains all the parameters that are estimated in a MODEL statement. The OUTEST= option can be used when the PROC MDC call contains one MODEL statement. There are additional restrictions. For the HEV and multinomial probit models, you need to specify exactly all possible elements of the choice set, since additional parameters (for example, SCALE1 or STD1) are generated automatically in the MDC procedure. Therefore, the following SAS statements are not valid when the OUTEST= option is speci ed:

```
proc mdc data=a outest=e;
    model y = x / type=hev choice=(alter);
run;
```

You need to specify all possible choices in the CHOICE= option since the OUTEST= option is speci ed as follows:

```
proc mdc data=a outest=e;
    model y = x / type=hev choice=(alter 1 2 3);
run;
```

When the NCHOICE= option is speci ed, no additional information about possible choices is required. Therefore, the following SAS statements are correct:

```
proc mdc data=a outest=e;
    model y = x / type=mprobit nchoice=3;
run;
```

The nested logit model does not produce the OUTEST= data set unless the NEST statement is speci ed.

Each parameter contains the estimate for the corresponding parameter in the corresponding model. In addition, the OUTEST= data set contains the following variables:

DEPVAR the name of the dependent variable

METHOD the estimation method

MODEL the label of the MODEL statement if one is speci ed, or blank otherwise

STATUS a character variable that indicates whether the optimization process reached convergence or failed to converge: 0 indicates that the convergence was reached, 1 indicates that the maximum number of iterations allowed was exceeded, 2 indicates a failure to improve the function value, and 3 indicates a failure to converge because the objective function or its derivatives could not be evaluated or improved, or linear constraints were dependent, or the algorithm failed to return to feasible region, or the number of iterations was greater than prespeci ed.

NAME the name of the row of the covariance matrix for the parameter estimate, if the COVOUT option is speci ed, or blank otherwise

LIKLHD the log-likelihood value

STDERR standard error of the parameter estimate, if the COVOUT option is speci ed

TYPE PARMS for observations that contain parameter estimates, or COV for observations that contain covariance matrix elements

The OUTEST= data set contains one observation for the MODEL statement giving the parameter estimates for that model. If the COVOUT option is speci ed, the OUTEST= data set includes additional observations for the MODEL statement giving the rows of the covariance matrix of parameter estimates. For covariance observations, the value of the _TYPE_ variable is COV, and the _NAME_ variable identi es the parameter associated with that row of the covariance matrix.

ODS Table Names

PROC MDC assigns a name to each table it creates. You can use these names to denote the table when using the Output Delivery System (ODS) to select tables and create output data sets. These names are listed in the Table 18.3.

Table 18.3 ODS Tables Produced in PROC MDC

ODS Table Name	Description	Option
ODS Tables Created by the MODEL Statement		
ResponsePro le	Response pro le	Default
ClassLevels	Class levels	Default
FitSummary	Summary of nonlinear estimation	Default
GoodnessOfFit	Pseudo-R-square measures	Default
ConvergenceStatus	Convergence status	Default
ParameterEstimates	Parameter estimates	Default
CovB	Covariance of parameter estimates	COVB
CorrB	Correlation of parameter estimates	CORRB
LinCon	Linear constraints	ITPRINT
InputOptions	Input options	ITPRINT
ProblemDescription	Problem description	ITPRINT
IterStart	Optimization start	ITPRINT
IterHist	Iteration history	ITPRINT
IterStop	Optimization results	ITPRINT
ConvergenceStatus	Convergence status	ITPRINT
ParameterEstimatesResults	Resulting parameters	ITPRINT
LinConSol	Linear constraints evaluated at solution	ITPRINT
ODS Tables Created by the TEST Statement		
TestResults	Test results	Default

Examples: MDC Procedure

Example 18.1: Binary Data Modeling

The MDC procedure supports various multinomial choice models. However, you can also use PROC MDC to estimate binary choice models such as binary logit and probit because these models are special cases of multinomial models.

Spector and Mazzeo (1980) studied the effectiveness of a new teaching method on students' performance in an economics course. They reported grade point average (gpa), previous knowledge of the material (tuce), a dummy variable for the new teaching method (psi), and the final course grade (grade). A value of 1 is recorded for grade if a student earned the letter grade "A," and 0 otherwise.

The binary logit can be estimated using the conditional logit model. In order to use the MDC procedure, the data are converted as follows so that each possible choice corresponds to one observation:

```
data smdata;
    input gpa tuce psi grade;
datalines;
2.66        20        0         0
2.89        22        0         0
3.28        24        0         0
2.92        12        0         0

    ... more lines ...


data smdata1;
    set smdata;
    retain id 0;
    id + 1;

    /*-- first choice --        */
    choice1 = 1;
    choice2 = 0;
    decision = (grade = 0);
    gpa_2 = 0;
    tuce_2 = 0;
    psi_2 = 0;
    output;

    /*-- second choice --       */
    choice1 = 0;
    choice2 = 1;
    decision = (grade = 1);
    gpa_2 = gpa;
    tuce_2 = tuce;
    psi_2 = psi;
    output;
run;
```

The first 10 observations are displayed in Output 18.1.1. The variables related to grade=0 are omitted since these are not used for binary choice model estimation.

Output 18.1.1 Converted Binary Data

Consider the choice probability of the conditional logit model for binary choice:

$$P_i(j) = \frac{\exp(x_{ij}'\beta)}{\sum_{k=1}^{2}\exp(x_{ik}'\beta)}; \quad j = 1, 2$$

The choice probability of the binary logit model is computed based on normalization. The preceding conditional logit model can be converted as

$$P_i(1) = \frac{1}{1 + \exp(-(x_{i2} - x_{i1})'\beta)}$$

$$P_i(2) = \frac{\exp(-(x_{i2} - x_{i1})'\beta)}{1 + \exp(-(x_{i2} - x_{i1})'\beta)}$$

Therefore, you can interpret the binary choice data as the difference between the first and second choice characteristics. In the following statements, it is assumed that $x_{i1} = 0$. The binary logit model is estimated and displayed in Output 18.1.2.

```
/* -- Conditional Logit --          */
proc mdc data=smdata1;
     model decision = choice2 gpa_2 tuce_2 psi_2 /
                      type=clogit
                      nchoice=2
                      covest=hess;
     id id;
run;
```

Output 18.1.2 Binary Logit Estimates

Consider the choice probability of the multinomial probit model:

$$P_i(j) = \Pr(\varepsilon_{i1} - \varepsilon_{ij} < (x_{ij} - x_{i1})'\beta, \ldots, \varepsilon_{iJ} - \varepsilon_{ij} < (x_{ij} - x_{iJ})'\beta)$$

The probabilities of choice of the two alternatives can be written as

$$P_i(1) = \Pr(\varepsilon_{i2} - \varepsilon_{i1} < (x_{i1} - x_{i2})'\beta)$$

$$P_i(2) = \Pr(\varepsilon_{i1} - \varepsilon_{i2} < (x_{i2} - x_{i1})'\beta)$$

where $\begin{pmatrix}\varepsilon_{i1}\\ \varepsilon_{i2}\end{pmatrix} \sim N\left(0, \begin{pmatrix}\sigma_1^2 & \sigma_{12}\\ \sigma_{12} & \sigma_2^2\end{pmatrix}\right)$. Assume that $x_{i1} = 0$ and $\sigma_{12} = 0$. The binary probit model is estimated and displayed in Output 18.1.3. You do not get the same estimates as that of the usual binary probit model. The probabilities of choice in the binary probit model are

$$P_i(2) = \Pr(\varepsilon_i < x_i'\beta)$$

$$P_i(1) = 1 - \Pr(\varepsilon_i < x_i'\beta)$$

where $\varepsilon_i \sim N(0, 1)$. However, the multinomial probit model has the error variance $\mathrm{Var}(\varepsilon_{i2} - \varepsilon_{i1}) = \sigma_1^2 + \sigma_2^2$ if ε_{i1} and ε_{i2} are independent ($\sigma_{12} = 0$). In the following statements, unit variance restrictions are imposed on choices 1 and 2 ($\sigma_1^2 = \sigma_2^2 = 1$). Therefore, the usual binary probit estimates (and standard errors) can be obtained by multiplying the multinomial probit estimates (and standard errors) in Output 18.1.3 by $1/\sqrt{2}$.

```
/*-- Multinomial Probit --*/
proc mdc data=smdata1;
    model decision = choice2 gpa_2 tuce_2 psi_2 /
                type=mprobit
                nchoice=2
                covest=hess
                unitvariance=(1 2);
    id id;
run;
```

Output 18.1.3 Binary Probit Estimates

Example 18.2: Conditional Logit and Data Conversion

In this example, data are prepared for use by the MDCDATA statement. Sometimes, choice-specific information is stored in multiple variables. Since the MDC procedure requires multiple observations for each decision maker, you need to arrange the data so that there is an observation for each subject-alternative (individual-choice) combination. Simple binary choice data are obtained from Ben-Akiva and Lerman (1985). The following statements create the SAS data set:

```
data travel;
    length mode $ 8;
    input auto transit mode $;
datalines;
52.9    4.4  Transit
4.1    28.5  Transit
4.1    86.9  Auto
56.2   31.6  Transit
51.8   20.2  Transit
0.2    91.2  Auto
27.6   79.7  Auto
89.9   2.2   Transit
41.5   24.5  Transit
95.0   43.5  Transit
99.1   8.4   Transit

    ... more lines ...
```

The travel time is stored in two variables, auto and transit. In addition, the chosen alternatives are stored in a character variable, mode. The choice variable, mode, is converted to a numeric variable, decision, since the MDC procedure supports only numeric variables. The following statements convert the original data set, travel, and estimate the binary logit model. The first 10 observations of a relevant subset of the new data set and the parameter estimates are displayed in Output 18.2.1 and Output 18.2.2, respectively.

```
data new;
    set travel;
    retain id 0;
    id+1;
    /* -- create auto variable -- */
```

```
    decision = (upcase(mode) = AUTO );
    ttime = auto;
    autodum = 1;
    trandum = 0;
    output;
    /*-- create transit variable --          */
    decision = (upcase(mode) = TRANSIT );
    ttime = transit;
    autodum = 0;
    trandum = 1;
    output;
run;

proc print data=new(obs=10);
    var decision autodum trandum ttime;
    id id;
run;
```

Output 18.2.1 Converted Data

The following statements perform the binary logit estimation:

```
proc mdc data=new;
    model decision = autodum ttime /
                type=clogit
                nchoice=2;
    id id;
run;
```

Output 18.2.2 Binary Logit Estimation of Modal Choice Data

In order to handle more general cases, you can use the MDCDATA statement. Choice-specific dummy variables are generated and multiple observations for each individual are created. The following example converts the original data set travel by using the MDCDATA statement and performs conditional logit analysis. Interleaved data are output into the new data set new3. This data set has twice as many observations as the original travel data set.

```
proc mdc data=travel;
    mdcdata varlist( x1 = (auto transit) )
                select=mode
                id=id
                alt=alternative
                decvar=Decision / out=new3;
    model decision = auto x1 /
                nchoice=2
                type=clogit;
    id id;
run;
```

The first nine observations of the modified data set are shown in Output 18.2.3. The result of the preceding program is listed in Output 18.2.4.

Output 18.2.3 Transformed Model Choice Data

Output 18.2.4 Results Using MDCDATA Statement

Example 18.3: Correlated Choice Modeling

Often, it is not realistic to assume that the random components of utility for all choices are independent. This example shows the solution to the problem of correlated random components by using multinomial probit and nested logit.

To analyze correlated data, trinomial choice data (1,000 observations) are created using a pseudo-random number generator by using the following statements. The random utility function is

$$U_{ij} = V_{ij} + \epsilon_{ij}; \quad j = 1, 2, 3$$

where

$$\epsilon_{ij} \sim N\left(0, \begin{bmatrix} 2 & .6 & 0 \\ .6 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}\right)$$

```
/*-- generate simulated series --          */
%let ndim = 3;
%let nobs = 1000;

data trichoice;
    array error{&ndim} e1-e3;
    array vtemp{&ndim} _temporary_;
    array lm{6} _temporary_ (1.4142136 0.4242641 1 0 0 1);
    retain nseed 345678;

    do id = 1 to &nobs;
        index = 0;
        /* generate independent normal variate          */
        do i = 1 to &ndim;
            /* index of diagonal element        */
            vtemp{i} = rannor(nseed);
        end;
        /* get multivariate normal variate             */
        index = 0;
        do i = 1 to &ndim;
            error{i} = 0;
            do j = 1 to i;
                error{i} = error{i} + lm{index+j}            *vtemp{j};
            end;
            index = index + i;
        end;
        x1 = 1.0 + 2.0   * ranuni(nseed);
        x2 = 1.2 + 2.0   * ranuni(nseed);
        x3 = 1.5 + 1.2   * ranuni(nseed);
        util1 = 2.0    * x1 + e1;
        util2 = 2.0    * x2 + e2;
        util3 = 2.0    * x3 + e3;
        do i = 1 to &ndim;
            vtemp{i} = 0;
        end;
        if ( util1 > util2 & util1 > util3 ) then
            vtemp{1} = 1;
        else if ( util2 > util1 & util2 > util3 ) then
            vtemp{2} = 1;
        else if ( util3 > util1 & util3 > util2 ) then
            vtemp{3} = 1;
        else continue;
        /*-- first choice --        */
```

```
        x = x1;
        mode = 1;
        decision = vtemp{1};
        output;
        /*-- second choice --*/
        x = x2;
        mode = 2;
        decision = vtemp{2};
        output;
        /*-- third choice --*/
        x = x3;
        mode = 3;
        decision = vtemp{3};
        output;
    end;
run;
```

First, the multinomial probit model is estimated (see the following statements). Results show that the standard deviation, correlation, and slope estimates are close to the parameter values. Note that $_{12} = \frac{_{12}}{\sqrt{_{1}^2}\sqrt{_{2}^2}} = \frac{0.6}{\sqrt{.2}\sqrt{.1}} = 0.42$, $_1 = \bar{\beta}_2 = 1.41$, $_2 = \bar{\beta}_1 = 1$, and the parameter value for the variable x is 2.0. (See Output 18.3.1.)

```
/*-- Trinomial Probit --*/
proc mdc data=trichoice randnum=halton nsimul=100;
    model decision = x /
                type=mprobit
                choice=(mode 1 2 3)
                covest=op
                optmethod=qn;
    id id;
run;
```

Output 18.3.1 Trinomial Probit Model Estimation

Figure 18.29 shows a two-level decision tree.

Figure 18.29 Nested Tree Structure

The following statements estimate the nested model shown in Figure 18.29:

```
/*-- Two-Level Nested Logit --        */
proc mdc data=trichoice;
    model decision = x /
                type=nlogit
                choice=(mode 1 2 3)
                covest=op
                optmethod=qn;
    id id;
    utility u(1,) = x;
    nest level(1) = (1 2 @ 1, 3 @ 2),
         level(2) = (1 2 @ 1);
run;
```

The estimated result (see Output 18.3.2) shows that the data support the nested tree model since the estimates of the inclusive value parameters are signi cant and are less than 1.

Output 18.3.2 Two-Level Nested Logit

Example 18.4: Testing for Homoscedasticity of the Utility Function

The conditional logit model imposes equal variances on random components of utility of all alternatives. This assumption can often be too restrictive and the calculated results misleading. This example shows several approaches to testing the homoscedasticity assumption.

The section “Getting Started: MDC Procedure” on page 991 analyzes an HEV model by using Daganzo's trinomial choice data and displays the HEV parameter estimates in Figure 18.15. The inverted scale estimates for mode “2” and mode “3” suggest that the conditional logit model (which imposes equal variances on random components of utility of all alternatives) might be misleading. The HEV estimation summary from that analysis is repeated in Output 18.4.1.

Output 18.4.1 HEV Estimation Summary ($_1$ D 1)

You can estimate the HEV model with unit scale restrictions on all three alternatives ($_1$ D $_2$ D $_3$ D 1) with the following statements.

```
/*-- HEV Estimation --      */
proc mdc data=newdata;
     model decision = ttime /
                      type=hev
                      nchoice=3
                      hev=(unitscale=1 2 3, integrate=laguerre)
                      covest=hess;
     id pid;
run;
```

Output 18.4.2 displays the estimation summary.

Output 18.4.2 HEV Estimation Summary ($_1$ D $_2$ D $_3$ D 1)

The test for scale equivalence (SCALE2=SCALE3=1) is performed using a likelihood ratio test statistic. The following SAS statements compute the test statistic (1.4276) and its p-value (0.4898) from the log-likelihood values in Output 18.4.1 and Output 18.4.2:

```
data _null_;
   /* -- test for H0: scale2 = scale3 = 1 -- */
   /*    ln L(max) = -34.1276                */
   /*    ln L(0)   = -33.4138                */
   stat = -2 * ( -34.1276 + 33.4138 );
   df = 2;
   p_value = 1 - probchi(stat, df);
   put stat= p_value=;
run;
```

The test statistic fails to reject the null hypothesis of equal scale parameters, which implies that the random utility function is homoscedastic.

A multinomial probit model also allows heteroscedasticity of the random components of utility for different alternatives. Consider the utility function

$$U_{ij} = V_{ij} + \epsilon_{ij}$$

where

$$\epsilon_i \sim N\left(0, \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \sigma_3^2 \end{bmatrix}\right)$$

This multinomial probit model is estimated by using the following statements:

```
/* -- Heteroscedastic Multinomial Probit -- */
proc mdc data=newdata;
   model decision = ttime /
                type=mprobit
                nchoice=3
                unitvariance=(1 2)
                covest=hess;
   id pid;
   restrict RHO_31 = 0;
run;
```

The estimation summary is displayed in Output 18.4.3.

Output 18.4.3 Heteroscedastic Multinomial Probit Estimation Summary

Next, the multinomial probit model with unit variances ($\sigma_1 = \sigma_2 = \sigma_3 = 1$) is estimated in the following statements:

```
/*-- Homoscedastic Multinomial Probit --                    */
proc mdc data=newdata;
     model decision = ttime /
                      type=mprobit
                      nchoice=3
                      unitvariance=(1 2 3)
                      covest=hess;
     id pid;
     restrict RHO_21 = 0;
run;
```

The estimation summary is displayed in Output 18.4.4.

Output 18.4.4 Homoscedastic Multinomial Probit Estimation Summary

The test for homoscedasticity ($\sigma_3 = 1$) under $\lambda_1 = \lambda_2 = 1$ shows that the error variance is not heteroscedastic since the test statistic (1.313) is less than $\chi^2_{0.05;1} = 3.84$. The marginal probability or p-value computed in the following statements from the PROBCHI function is 0.2519:

```
data _null_;
   /* -- test for H0: sigma3 = 1 --        */
   /*      ln L(max) = -33.8860             */
   /*      ln L(0)   = -34.5425             */
   stat = -2 * ( -34.5425 + 33.8860 );
   df = 1;
   p_value = 1 - probchi(stat, df);
   put stat= p_value=;
run;
```

Example 18.5: Choice of Time for Work Trips: Nested Logit Analysis

This example uses sample data of 527 automobile commuters in the San Francisco Bay Area to demonstrate the use of nested logit model.

Brownstone and Small (1989) analyzed a two-level nested logit model that is displayed in Figure 18.30. The probability of choosing j at level 2 is written as

$$P_i(j) = \frac{\exp(\lambda_j I_j)}{\sum_{j'=1}^{3} \exp(\lambda_{j'} I_{j'})}$$

where $I_{j'}$ is an inclusive value and is computed as

$$I_{j'} = \ln\left[\sum_{k' \in C_{j'}} \exp(x'_{ik'}\beta)\right]$$

The probability of choosing an alternative k is denoted as

$$P_i(k|j) = \frac{\exp(x'_{ik}\beta)}{\sum_{k' \in C_j} \exp(x'_{ik'}\beta)}$$

The full information maximum likelihood (FIML) method maximizes the following log-likelihood function:

$$L = \sum_{i=1}^{N} \sum_{j=1}^{J} d_{ij} \{\ln[P_i(k|j)] + \ln[P_i(j)]\}$$

where $d_{ij} = 1$ if a decision maker i chooses j, and 0 otherwise.

Figure 18.30 Decision Tree for Two-Level Nested Logit

Sample data of 527 automobile commuters in the San Francisco Bay Area have been analyzed by Small (1982); Brownstone and Small (1989). The regular time of arrival is recorded as between 42.5 minutes early and 17.5 minutes late, and indexed by 12 alternatives, using ve-minute interval groups. Refer to Small (1982) for more details on these data. The following statements estimate the two-level nested logit model:

```
/*-- Two-level Nested Logit --          */
proc mdc data=small maxit=200 outest=a;
     model decision = r15 r10 ttime ttime_cp sde sde_cp
                              sdl sdlx d2l /
                 type=nlogit
                 choice=(alt);
     id id;
     utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                              sdl sdlx d2l;
     nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
          level(2) = (1 2 3 @ 1);
run;
```

The following statements add the upalt variable, which describes the choice at the upper level of the nested tree to the data set.

```
data small;
     set small;
     upalt=1;
     if alt=9 then upalt=2;
     if alt>9 then upalt=3;
run;
```

The following statements show and alternative speci cation, which uses the CHOICE= option with two nested levels that are represented by upalt and alt:

```
proc mdc data=upalt maxit=200;
    model decision = r15 r10 ttime ttime_cp sde sde_cp
                             sdl sdlx d2l /
                type=nlogit
                choice=(upalt,alt);
```

```
    id id;
    utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                     sdl sdlx d2l;
run;
```

The estimation summary, discrete response pro le, and the FIML estimates are displayed in Output 18.5.1 through Output 18.5.3.

Output 18.5.1 Nested Logit Estimation Summary

Output 18.5.2 Discrete Choice Characteristics

Output 18.5.3 Nested Logit Estimates

Now policy makers are particularly interested in predicting shares of each alternative to be chosen by population. One application of such predictions are market shares. Going even further, it is extremely useful to predict choice probabilities out of sample; that is, under alternative policies.

Suppose that in this particular transportation example you are interested in projecting the effect of a new program that indirectly shifts individual preferences with respect to late arrival to work. This means that you manage to decrease the coef cient for the “late dummy” D2L, which is a penalty for violating some margin of arriving on time. Suppose that you alter it from an estimated 1:0941 to almost twice that level, 2:0941

But rst, in order to have a benchmark share, you predict probabilities to choose each particular option and output them to the new data set with the following additional statement:

```
/* -- Create new data set with predicted probabilities -- */
output out=predicted1 p=probs;
```

Having these in sample predictions, you sort the data by alternative and aggregate across each of them as shown in the following statements:

```
/* -- Sort the data by alternative -- */
proc sort data=predicted1;
   by alt;
run;

/* -- Calculate average probabilities of each alternative -- */
proc means data=predicted1 nonobs mean;
   var probs;
   class alt;
run;
```

Output 18.5.4 shows the summary table that is produced by the preceding statements.

Output 18.5.4 Average Probabilities of Choosing Each Particular Alternative

Now you change the preference parameter for variable D2L. In order to x all the parameters, you use the MAXIT=0 option to prevent optimization and the START= option in MODEL statement to specify initial parameters.

```
/*-- Two-level Nested Logit --          */
proc mdc data=small maxit=0 outest=a;
    model decision = r15 r10 ttime ttime_cp sde sde_cp
                     sdl sdlx d2l /
                type=nlogit
                choice=(alt)
                start=( 1.1034 0.3931 -0.0465 -0.0498
                        -0.6618 0.0519 -2.1006 -3.5240
                        -2.0941 0.6762  1.0906  0.7622);
    id id;
    utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                     sdl sdlx d2l;
    nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
         level(2) = (1 2 3 @ 1);
    output out=predicted2 p=probs;
run;
```

You apply the same SORT and MEANS procedures as applied earlier to obtain the following summary table in Output 18.5.5.

Output 18.5.5 Average Probabilities of Choosing Each Particular Alternative after Changing the Preference Parameter

Comparing the two tables shown in Output 18.5.4 and Output 18.5.5, you clearly see the effect of increased dislike of late arrival. People shifted their choices towards earlier times (alternatives 1–8) from the on-time option (alternative 9).

Brownstone and Small (1989) also estimate the two-level nested logit model with equal scale parameter constraints, $_1$ D $_2$ D $_3$. Replication of their model estimation is shown in the following statements:

```
/*-- Nested Logit with Equal Dissimilarity Parameters --                    */
proc mdc data=small maxit=200 outest=a;
    model decision = r15 r10 ttime ttime_cp sde sde_cp
                             sdl sdlx d2l /
                samescale
                type=nlogit
                choice=(alt);
    id id;
    utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                             sdl sdlx d2l;
    nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
         level(2) = (1 2 3 @ 1);
run;
```

The parameter estimates and standard errors are almost identical to those in Brownstone and Small (1989, p. 69). Output 18.5.6 and Output 18.5.7 display the results.

Output 18.5.6 Nested Logit Estimation Summary with Equal Dissimilarity Parameters

Output 18.5.7 Nested Logit Estimates with Equal Dissimilarity Parameters

However, the test statistic for $H_0: \tau_1 = \tau_2 = \tau_3$ rejects the null hypothesis at the 5% significance level since $-2 \cdot \ln L(0) - \ln L(\hat{\theta}) = 7.15 > \chi^2_{0.05;2} = 5.99$. The p-value is computed in the following statements and is equal to 0.0280:

```
data _null_;
   /* -- test for H0: tau1 = tau2 = tau3 -- */
   /*    ln L(max) = -990.8191              */
   /*    ln L(0)   = -994.3940              */
   stat = -2 * ( -994.3940 + 990.8191 );
   df = 2;
   p_value = 1 - probchi(stat, df);
   put stat= p_value=;
run;
```

Example 18.6: Hausman's Speci cation Test

As discussed under multinomial and conditional logits, the odds ratios in the multinomial or conditional logits are independent of the other alternatives. See the section "Multinomial Logit and Conditional Logit" on page 1024. This property of the logit models is often viewed as rather restrictive and provides substitution patterns that do not represent the actual relationship among choice alternatives.

This independence assumption, called independence of irrelevant alternatives (IIA), can be tested with Hausman's speci cation test. According to Hausman and McFadden (1984), if a subset of choice alternatives is irrelevant, it can be omitted from the sample without changing the remaining parameters systematically.

Under the null hypothesis (IIA holds), omitting the irrelevant alternatives leads to consistent and ef cient parameter estimates β_R, while parameter estimates β_U from the unrestricted model are consistent but inef cient. Under the alternative, only the parameter estimates obtained from the unrestricted model are consistent.

This example demonstrates the use of Hausman's speci cation test to analyze the IIA assumption and decide on an appropriate model that provides less restrictive substitution patterns (nested logit or multinomial probit). A sample data set of 527 automobile commuters in the San Francisco Bay Area is used (Small 1982). The regular time of arrival is recorded as between 42.5 minutes early and 17.5 minutes late, and is indexed by 12 alternatives, using ve-minute interval groups. See Small (1982) for more details about these data.

The data can be divided into three groups: commuters who arrive early (alternatives 1 – 8), commuters who arrive on time (alternative 9), and commuters who arrive late (alternatives 10 – 12). Suppose that you want to test whether the IIA assumption holds for commuters who arrived on time (alternative 9).

Hausman's speci cation test is distributed as χ^2 with k degrees of freedom (equal to the number of independent variables) and can be written as

$$\chi^2 = (\beta_U - \beta_R)' [V_R - V_U]^{-1} (\beta_U - \beta_R)$$

where β_R and V_R represent parameter estimates and the variance-covariance matrix, respectively, from the model where the ninth alternative was omitted, and β_U and V_U represent parameter estimates and the variance-covariance matrix, respectively, from the full model. The following macro can be used to perform the IIA test for the ninth alternative:

```
/* ------------------------------------------------------------------
 * name: %IIA
 * note: This macro test the IIA hypothesis using the Hausman s
 *       specification test. Inputs into the macro are as follows:
 *       indata:     input data set
 *       varlist:    list of RHS variables
 *       nchoice:    number of choices for each individual
 *       choice:     list of choices
 *       nvar:       number of independent variables
 *       nIIA:       number of choice alternatives used to test IIA
 *       IIA:        choice alternatives used to test IIA
 *       id:         ID variable
 *       decision:   0-1 LHS variable representing nchoice choices
 * purpose: Hausman s specification test
```

```
 * ------------------------------------------------------------------ */

%macro IIA(indata=, varlist=, nchoice=, choice= , nvar= , IIA= ,
                    nIIA=, id= , decision=);

    %let n=%eval(&nchoice-&nIIA);

    proc mdc data=&indata outest=cov covout ;
        model &decision = &varlist /
                    type=clogit
                    nchoice=&nchoice;
        id &id;
    run;

    data two;
        set &indata;
        if &choice in &IIA and &decision=1 then output;
    run;

    data two;
        set two;
        keep &id ind;
        ind=1;
    run;

    data merged;
        merge &indata two;
        by &id;
        if ind=1 or &choice in &IIA then delete;
    run;

    proc mdc data=merged outest=cov2 covout ;
        model &decision = &varlist /
                    type=clogit
                    nchoice=&n;
        id &id;
    run;

    proc IML;
        use cov var{_TYPE_ &varlist};
            read first into BetaU;
            read all into CovVarU where(_TYPE_= COV );
        close cov;

        use cov2 var{_TYPE_ &varlist};
            read first into BetaR;
            read all into CovVarR where(_TYPE_= COV );
        close cov;

        tmp = BetaU-BetaR;
        ChiSq=tmp * ginv(CovVarR-CovVarU) * tmp ;
        if ChiSq<0 then ChiSq=0;
        Prob=1-Probchi(ChiSq, &nvar);
        Print "Hausman Test for IIA for Variable &IIA";
```

```
        Print ChiSq Prob;
    run; quit;

%mend IIA;
```

The following statement invokes the %IIA macro to test IIA for commuters who arrive on time:

```
%IIA( indata=small,
      varlist=r15 r10 ttime ttime_cp sde sde_cp sdl sdlx d2l,
      nchoice=12,
      choice=alt,
      nvar=9,
      nIIA=1,
      IIA=(9),
      id=id,
      decision=decision );
```

The obtained ^{2} of 7.9 and the p-value of 0.54 indicate that IIA holds for commuters who arrive on time (alternative 9). If the IIA assumption did not hold, the following model (nested logit), which reserves a subcategory for alternative 9, might be more appropriate. See Output 18.30.

```
proc mdc data=small maxit=200 outest=a;
    model decision = r15 r10 ttime ttime_cp sde sde_cp
                     sdl sdlx d2l /
             type=nlogit
             choice=(alt);
    id id;
    utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                     sdl sdlx d2l;
    nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
         level(2) = (1 2 3 @ 1);
run;
```

Similarly, IIA could be tested for commuters who arrive approximately on time (alternative 8, 9, 10), as follows:

```
%IIA( indata=small,
      varlist=r15 r10 ttime ttime_cp sde sde_cp sdl sdlx d2l,
      nchoice=12,
      choice=alt,
      nvar=9,
      nIIA=3,
      IIA=(8 9 10),
      id=id,
      decision=decision );
```

Based on this test, independence of irrelevant alternatives is not rejected for this subgroup (^{2} = 3.3 and p-value=0.326), and it is concluded that a more complex nested logit model with commuters who arrive approximately on time in one subcategory is not needed. Since the two Hausman's speci cation tests just performed did not reject IIA, it might be a good idea to test whether the nested logit model is even needed. This is done using the likelihood ratio test in the next example.

Example 18.7: Likelihood Ratio Test

This example is an extension of Example 18.6; it performs another speci cation test, the likelihood ratio test (LR). Suppose you are interested in testing whether the nested logit model (Output 18.30) with three subgroups that represent commuters who arrive early, on time, and late is more appropriate than the standard multinomial logit. This can be done by adding the TEST statement to the model as follows:

```
/* -- Restricted Model with Inclusive Value Parameters
      Constrained to One --      */
proc mdc data=small maxit=200 outest=a;
    model decision = r15 r10 ttime ttime_cp sde sde_cp
                     sdl sdlx d2l /
                type=nlogit
                choice=(alt);
    id id;
    utility u(1, ) = r15 r10 ttime ttime_cp sde sde_cp
                     sdl sdlx d2l;
    nest level(1) = (1 2 3 4 5 6 7 8 @ 1, 9 @ 2, 10 11 12 @ 3),
         level(2) = (1 2 3 @ 1);
    test INC_L2G1C1=1, INC_L2G1C2=1, INC_L2G1C3=1 /LR;
run;
```

Output 18.7.1 Likelihood Ratio Test

Based on this test, you can conclude that the inclusive values INC_L2G1C1, INC_L2G1C2, and INC_L2G1C3 are jointly statistically different from the value 1 at the 5% level and therefore the nested logit is a more appropriate model. The LR test can be used to test other types of restrictions in the nested logit setting as long as one model can be nested within another.

Acknowledgments: MDC Procedure

Professor Kenneth Small provided the work trip data that are used in the “Examples” section. These data were collected for the urban travel demand forecasting project, which was carried out by McFadden, Talvitie, and Associates (1977). The project was supported by the National Science Foundation, Research Applied to National Needs Program through grants GI-43740 and APR74-20392, and the Alfred P. Sloan Foundation, through grant 74-21-8.