Estimation of Discrete Choice Models Using DCM for Ox

Matias Eklöf and Melvyn Weeks

April 2004

CWPE 0427

Not to be quoted without permission

Abstract

DCM (Discrete Choice Models) is a package for estimating a class of discrete choice models. Written in Ox, DCM is a class that implements a wide range of discrete choice models including standard binary response models, with notable extensions including conditional mixed logit, mixed probit, multinomial probit, and random coefficient ordered choice models. The current version can handle both cross-section and static panel data. DCM represents an important development for the discrete choice computing environment in making available a broad range of models which are now widely used by academics and practitioners. Developed as a derived class of `Modelbase`, users may access the functions within DCM by either writing Ox programs which create and use an object of the DCM class, or use the program in an interactive fashion via OxPack in GiveWin. We demonstrate the capabilities of DCM by using a number of applications from the discrete choice literature.

JEL Classification: C87, C25, C15

Keywords: Discrete choice models, simulation methods, multinomial probit, mixed logit, ordinal response, revealed preference.

Estimation of Discrete Choice Models Using DCM for Ox

Matias Eklöf,
Department of Economics, Uppsala University,
Sweden.
Melvyn Weeks
University of Cambridge

1 Introduction

DCM (Discrete Choice Models) is a package for estimating a class of discrete choice models. Written in Ox, DCM is a class that implements a wide range of discrete choice models including standard binary response models, with notable extensions including conditional mixed logit, mixed probit, multinomial probit, and random coefficient ordered choice models. The current version can handle both cross-section and static panel data. DCM represents an important development for the discrete choice computing environment in making available a broad range of models which are now widely used by academics and practitioners. Developed as a derived class of `Modelbase`, users may access the functions within DCM by either writing Ox programs which create and use an object of the DCM class, or use the program in an interactive fashion via OxPack in GiveWin. We demonstrate the capabilities of DCM by using a number of applications from the discrete choice literature.

In this paper we outline the functionality of a new piece of software for estimation in discrete choice models, and demonstrate its use with reference to a number of applications. In section 2 we introduce notation and in order to provide the user with the necessary econometric background, we provide an overview of a broad class of multiple and single index discrete choice models. In section 3 and 4, respectively, we introduce the members of the multiple and single index class of models that are available in DCM. In section 5 we introduce the DCM class, and examine the main member functions. In section 6 we demonstrate the use of DCM by considering a number of applications.

1.1 Disclaimer

The DCM package is functional, but we provide no warranty. For general issues relating to Ox and the DCM package we refer users to the ox-users discussion group[1]. We are happy to receive suggestions for improvement, although the program for future updates and revisions will be determined by the authors. For Matias Eklof: matias.eklof@nek.uu.se; for Melvyn Weeks: Melvyn.Weeks@econ.cam.ac.uk.

1.2 Availability and Citation

DCM is available free of charge for academic users from
`http://www.econ.cam.ac.uk/faculty/weeks/DCM/`.

The only condition of use is that authors cite this document in all reports and publications involving the application of the DCM package.

1.3 Installation

1. Make sure you have properly installed Ox version 3.10 or later. The DCM package does not work fully with earlier versions of Ox. Type `oxl` at the command prompt to check.

2. Create a DCM subdirectory in the `ox/packages` folder and put `dcm100.zip` in that subdirectory, then unzip this file.

3. Read the `readme.txt` file for information on last minute changes.

4. If Ox has been installed properly, this will allow using the `DCM` package from any directory. To use the package in your code, add the command

   ```
   #include ''packages/dcm/dcm.ox''
   ```

   at the top of all files which require it.

[1] Subscription information and archiving is available at `www.mailbase.ac.uk/lists/ox-users`

1.4 Running DCM code

The current version of DCM runs under either the Ox Console version or GiveWin with OxProfessional. This implies that the user can write short Ox programs that creates the DCM object. In addition, there is an OxPack graphical user interface implemented where the user may run DCM interactively. OxEdit, an easy to use text editor which supports syntax highlighting and running external tools, may be used to both develop and run Ox programs. This editor can be obtained from `http://www.oxedit.com/oxedit.html`. A set of example files can also be downloaded from the DCM website.

2 Specification of Discrete Choice Models: A Canonical Framework

To introduce notation we consider the following general canonical model of discrete choice behaviour which includes both *individual specific characteristics* and *alternative specific attributes*.[2] For all that follows we assume that the investigator has knowledge of the density from which the observed sample is obtained and requires estimates of an unknown parameter vector, $\boldsymbol{\theta}$. The model is general in the sense of nesting a range of alternate formulations based upon the inclusion or exclusion of individual characteristics and alternative attributes, and allowing elements of $\boldsymbol{\theta}$ to be either fixed or random. For the sake of completeness we also consider discrete choice models that are both additively separable in these two types of variables, and those which utilise interaction effects. For example, in specifying a model which includes only individual characteristics, identification of model parameters is achieved by ascribing a parameter vector which varies over alternatives. A different (and perhaps more realistic) solution to the identification problem would be to allow individual characteristics to interact with alternative attributes. For example, if one believes that the

[2]Henceforth, variables that are alternative specific are referred to as "attributes", whereas variables that are individual specific (and thus invariant across alternatives) are referred to as "characteristics".

utility derived from the size of an automobile is dependent upon household size, then it would be necessary to include an interaction term representing this effect.

All individuals face a finite set, Ω_J, of discrete alternatives indexed by j. We employ a latent variable formulation where for a given individual the value (or utility) from alternative j may be expressed as

$$y_j^* = \alpha_j + \mathbf{x}'\boldsymbol{\beta}_j + \mathbf{v}_j'\boldsymbol{\omega} + \tau_{PI}(v_{jP}x_I) + \varepsilon_j, \qquad j = 1, ..., J. \tag{1}$$

We differentiate between the following components of $\boldsymbol{\theta}$: a $K \times 1$ vector $\boldsymbol{\beta}_j = \{\beta_{jk}\}$, containing parameters representing the effects of *individual characteristics* upon choice; a $L \times 1$ vector $\boldsymbol{\omega} = \{\omega_l\}$, denoting the effect of *alternative specific attributes*; and a $J \times 1$ vector $\boldsymbol{\alpha} = \{\alpha_j\}$ of alternative-specific constants. $\mathbf{x} = \{x_k\}$ is a $K \times 1$ vector of non-stochastic components of utility, containing the K alternative invariant individual characteristics of alternative j; $\mathbf{v}_j = \{v_{jl}\}$ is a $L \times 1$ vector of alternative specific attributes. $(v_{jP}x_I)$ represents an interaction term allowing for, as an example, the marginal evaluation of attribute v_P to vary according to an individual characteristic x_I; τ_{PI} is the associated coefficient.

The stochastic component of the model $\boldsymbol{\varepsilon} = \{\varepsilon_j\}$ is a $J \times 1$ vector of disturbance terms whose distribution is known, possibly up to a knowledge of a further set of unknown variance and covariance parameter. Writing $\boldsymbol{\varepsilon} \sim (\mathbf{0}, \Xi_\varepsilon)$, where Ξ_ε is the $J \times J$ covariance matrix of disturbance terms, we collect any free variance and covariance parameters in the vector $\boldsymbol{\kappa}$. (1) may be compactly written

$$\mathbf{y}^* = \boldsymbol{\alpha} + \tilde{\boldsymbol{\beta}}'\mathbf{x} + \mathbf{V}'\boldsymbol{\omega} + \tau_{PI}(\mathbf{v}_P \odot x_I) + \boldsymbol{\varepsilon} = \mathbf{D} + \boldsymbol{\varepsilon}, \tag{2}$$

where $\mathbf{y}^* = \{y_j^*\}$ collects the $J \times 1$ vector of utilities, $\tilde{\boldsymbol{\beta}}$ is a $K \times J$ vector, and $\mathbf{V}$ is a $L \times J$ matrix of alternative specific attributes, $\mathbf{v}_P$ is a $J \times 1$ vector, representing the J elements of the attribute with a marginal effect which varies according to the characteristic x_I. $\odot$ is the component by component product. $\mathbf{D} = \{D_j\} = \{\alpha_j + \mathbf{x}'\boldsymbol{\beta}_j + \mathbf{v}_j'\boldsymbol{\omega} + \tau_{PI}(v_{jP}x_I)\}$ denotes the deterministic component of choice, which we will also refer to as the linear index.

In what follows $i = 1, ..., N$ indexes individuals. At this juncture we implicitly assume that there are $T = 1$ choice occasions.

The combination of the $J \times 1$ linear index $\mathbf{D}$, assumptions regarding the stochastic component of choice $\boldsymbol{\varepsilon}$, an observational rule, and a link function $F(\mathbf{D})$, generate a class of multinomial response models. Ee note that these models are also referred to as multiple index models. DCM can also estimate single index models, such as ordered probit, and ordered mixed probit. By imposing an ordinality assumption on the choice set, single index models, such as ordinal response (OR) models, are able to circumvent dimensionality problems that arise in certain multiple index models. For example, the observational rule associated with OR models, may be written

$$y_i = \mathbf{1}(\alpha_{j-1} < y_i^* = D_i + \varepsilon_i < \alpha_j) \cdot j,$$

where α_{j-1} and α_j are unknown threshold parameters. For each individual D_i is a scalar, giving rise to the notion of a *single* index.

We note that in the current version there are limitations to the class of models available within DCM. First, although the class of models we consider is inherently non-linear, DCM can only handle models in which the index function $\mathbf{D}$ is linear. Second, in the discussion that follows kernel logit, mixed logit, and random coefficient logit are, in general, synonymous terms used to refer to an extension of a vanilla logit model encompassing a composite error structure permitting two additive components: a type I extreme value error, and an error component that facilitates a departure from iid disturbances due to random parameters. In this sense our canonical form is limited in the range of departures we consider from vanilla logit that maintain a logit kernel. For example, Ben-Akiva, Bolduc, and Walker (2001) refer to a highly general factor analytic logit kernel model which encompass a range of non iid variants, and include heteroscedastic logit, nested and cross-nested logit, together with a random parameter specification. In the current version of DCM departures from vanilla logit that maintain a logit kernel are the nested logit and the random parameters logit model.

3 Multiple Index Models

Given that $\mathbf{y}^*$ is unobserved, we require a mapping to represent the relationship between $\mathbf{y}^*$ and an observed outcome, say $\mathbf{y}$. We represent this relationship by the many-to-one observational rule

$$y = \kappa(\mathbf{y}^*), \tag{3}$$

where y is the index of the maximum element of the vector $\mathbf{y}^*$. Thus, in the case of multiple index discrete choice models the function $\kappa(\cdot)$ simply represents the *maximum index* of the components of $\mathbf{y}^*$. A given individual chooses alternative j' if the following set of linear inequality constraints are satisfied

$$\begin{gathered} -\infty < y_j^* < \infty \\ 0 < y_{j'}^* - y_j^* < \infty \forall j \neq j' \in \Omega_J. \end{gathered} \tag{4}$$

We begin by considering the most general density for $\boldsymbol{\varepsilon}$ in the form of the multivariate normal distribution, $\boldsymbol{\varepsilon} \sim MVN(0, \boldsymbol{\Xi}_\varepsilon)$.

3.1 Multinomial Probit

We write the conditional probability of choosing alternative j' as

$$\Pr(y = j'|\mathbf{z}, \boldsymbol{\theta}) = \int_{-\infty}^{(D_{j'}-D_1)} \ldots \int_{-\infty}^{(D_{j'}-D_J)} g(\eta_{1j'}, ..., \eta_{Jj'}, \Xi_{\varepsilon J-1}) d\eta_{1j'}, ..., \eta_{Jj'} \tag{5}$$

where $g(.)$ is a multivariate normal density of dimension $J-1$, with components $\eta_{sj} = \varepsilon_s - \varepsilon_j \; \forall s = 1, ..., J \; (s \neq j)$, $\mathbf{z}$ is the information set available to the analyst comprised of the observed attributes of alternatives and individual characteristics, and $\Xi_{\varepsilon J-1}$ is the covariance matrix for the error differences η_{sj}. Obviously as the dimension of Ω_J increases a curse of dimensionality makes the estimation of probability expressions such as (5) extremely time consuming. However, although much has been written on methods to circumvent the dimensionality problem, the problem of identification is also noteworthy, and logically precedes estimation.[3] Although this issue is covered in depth in an accompanying paper (see Eklof and Weeks (2003)), here we simply make a number of observations. First, using the observational rule in (3), if we consider the discrete information as imperfect measures on an underlying (latent) utility model, then the realisation that the analyst will only observe the sign of $U(j) - U(j') \ \forall \ j \neq j' \in \Omega$, will have implications for the identification of the location and scale of the model. Second, and following Ben-Akiva, Bolduc, and Walker (2001), DCM facilitates the specification of covariance matrix components that are alternative specific, and those that, in the case of a random parameter specification, vary over individuals. For example, in specifying a utility model with a representative agent determining the mean, we may consider a stochastic term as composed of two components: unobserved *attributes* of alternatives, and in the case of *observed characteristics*, deviations of individual tastes from an average.

Here we consider the following propositions for identification, based upon a number of alternative representations of the the random component ε_{ij}. See Eklof and Weeks (2003) for a proof of these propositions.

3.1.1 Identification under Different Covariance Structures

Proposition 1 (Alternative Specific Random Component) *For the multinomial probit model with J alternatives and an alternative-specific error covariance matrix Ξ_ε, and assuming that there is no random taste variation on observed attributes, a maximum of $(J-1)J/2-1$ free error covariance parameters are identified.*

Proposition 2 (Individual Specific Random Components) *For the multinomial probit model with J alternatives and random taste variation over all L attributes, then assuming that alternative-*

[3] Testimony to this observation has been provided by a number of authors including Bunch and Kitamura (1989) and Ben-Akiva, Bolduc, and Walker (2001) who note that in a number of published (and refereed) articles and textbooks, models were formally underidentified.

specific errors are iid, a maximum of $L(L+1)/2$ *covariance parameters are identified.*[4]

Proposition 3 (Individual and alternative specific random components) *For the multinomial probit model with J alternatives, with random taste variation over the L attributes and an alternative-specific error covariance matrix* $\Xi_{\varepsilon J}$*, a maximum of* $(J-1)J/2-1+(L+1)L/2$ *error covariance parameters are identified.*[5]

3.2 Approximating the MNP Model

The analytical tractability of (5) is critically dependent upon the restrictions placed upon $\Xi_{\varepsilon J-1.}$ The MNP model belongs to a class of models where both the criterion function and first order conditions are without a simple analytical form. In this particular case the maximisation of the likelihood function requires the evaluation of a multi-dimensional integral for each sample point. For choice problems where the dimension of Ω_J exceeds four the evaluation of multidimensional integrals is computationally prohibitive, hence the curse of dimensionality (see Bellman (1957)). In this case DCM provides the user with a number of options:

i) maintain an assumption of multivariate normality and utilise simulation techniques to circumvent the dimensionality constraint. See, for example, McFadden (1989) and Pakes and Pollard (1989).

ii) replace the stochastic specification $\boldsymbol{\varepsilon} \sim MVN(0, \Xi_\varepsilon)$ with $\boldsymbol{\varepsilon} \sim \Lambda(0,\gamma)$, where Λ denotes the Type 1 extreme value density

$$f(\varepsilon) = e^{-\varepsilon}e^{-e^{-\varepsilon}},$$

with variance terms $\gamma = \pi^2/6$. Dependent upon the mix of alternative specific attributes and individual-specific characteristics, we have the conditional and multinomial logit model.

[4]We note that the MNP model (implied in the above proposition) with Ξ_ε set equal to I_J is also referred to as the mixed probit model.

[5]This assumes that the scaling restriction has been set by fixing one of the elements of Ξ_ε.

iii) *mix* a standard logit model with an assumed distribution for the vector of mean parameters. In other words take a standard logit choice probability conditional upon a set of mean parameters, say $\boldsymbol{\beta}$, and weight each probability (at the level of the individual) with a weight given by the mixing density $f(\boldsymbol{\beta}|\boldsymbol{\eta})$, where $\boldsymbol{\eta}$ is a vector of hyperparameters. The resultant mixed logit model (MXL) may then be motivated using a random coefficient interpretation of RUM.

iv) The mixed logit model represents one way to circumvent the restrictive set of assumptions consistent with vanilla logit, whilst avoiding the dimensionality problems attendant with the MNP model. An alternative to this approach is the Nested Logit model, which is a member of a broad class of models based upon Generalised Extreme Value. The specific generalisation here is based upon the use of prior information to group the choice set into a number of mutually exclusive partitions, and imbue each partition with a common factor. The common factor, shared across members of the same partition, facilitates a departure from the IIA assumption *within* nests; and an IIN (independence of irrelevant nests) assumption across partitions. Below, we examine this form, and also demonstrate how the canonical form of the mixed logit model can be used to generate an analog nested logit model.

3.3 The Conditional and Multinomial Logit Model

The combination of simulation technology and computational power has elevated the MNP model to a feasible model of discrete choice. However, the logit model remains a highly tractable benchmark model across many users and applications. Writing the utility of choice j for individual i as

$$y_{ij}^{*} = D_{ij} + \varepsilon_{ij},$$

then for $\boldsymbol{\varepsilon}_i = (\varepsilon_{i1}, \varepsilon_{i2}, ..., \varepsilon_{iJ})'$ distributed independent and identically type I extreme value, the probability that individual i chooses

j may be written

$$\Pr(y_i = j|\mathbf{z}, \boldsymbol{\theta}) = \frac{e^{D_{ij}}}{\Sigma_{j=1}^{J} e^{D_{ij}}} = \frac{e^{\alpha_j + \mathbf{x}_i' \boldsymbol{\beta}_j} e^{\mathbf{v}_{ij}' \boldsymbol{\omega}}}{\Sigma_{j=1}^{J} e^{\alpha_j + \mathbf{x}_i' \boldsymbol{\beta}_j} e^{\mathbf{v}_{ij}' \boldsymbol{\omega}}}. \tag{6}$$

We note that (6) represents the probabilities for the *conditional* logit model. Note that D_{ij} contains both alternative specific attributes (in $\mathbf{v}_{ij}$) and individual characteristics (in $\mathbf{x}_i$) which are invariant over the choice set. Although the form of (6) is common in models of transportation choice, marketing, and also *stated* preference, most datasets analyzed by economists in models of *revealed* preference do not include alternative specific attributes. A model including only individual characteristics $\mathbf{x}_i$ is referred to as the *multinomial logit* model.

3.4 The Mixed Logit Model

We consider the simple the linear random utility model

$$\mathbf{y}_i^* = \boldsymbol{\alpha} + \mathbf{v}_i' \boldsymbol{\omega}_i + \boldsymbol{\varepsilon}_i, \tag{7}$$

where the parameter vector $\boldsymbol{\omega}_i = \bar{\boldsymbol{\omega}} + \tilde{\boldsymbol{\omega}}_i$ is a random quantity, with $\bar{\boldsymbol{\omega}}$ ($\tilde{\boldsymbol{\omega}}_i$) denoting a mean (individual specific) component. There are a number of distinguishing features concerning the mixed logit representation of random taste variation.[6] First, we write the distribution of $\boldsymbol{\omega}_i$ more generally as $\boldsymbol{\omega}_i \sim \Upsilon(\bar{\boldsymbol{\omega}}, \Xi_{\varpi})$, where Υ denotes a multivariate density with mean $\bar{\boldsymbol{\omega}}$ and covariance matrix Ξ_{ϖ}. Unlike the MNP model Υ need not be multivariate normal. Second, the $J \times 1$ vector of residual error terms $\boldsymbol{\varepsilon}_i$ is distributed *iid* Type 1 extreme value. This will have a number of implications, the

[6]Note that in this representation of the mixed logit model we choose to specify a mean component of utility which includes only a vector of alternative specific attributes. We do this to simplify notation. It is of course possible to estimate a model where functions of attributes are included, based upon interacting attributes and individual characteristics. In addition we present a model where mixing is imposed across all elements of $\mathbf{v}$. This need not be the case as discussed below.

most critical being that conditional on integrating out the random quantity $\boldsymbol{\omega}_i$, we are left with a logit probability (logit kernel). As a result, there is no need to impose any additional scale normalisations as, for example, is required in specifiying an estimable set of error covariance parameters in Ξ_ε in the case of multinomial probit.

3.4.1 The Mixed Logit Model: Choice Probabilities

Assuming a single observation per individual, probabilities for the mixed logit model (MXL) may be written

$$P_\Omega(y_i = j|\mathbf{v}, \boldsymbol{\eta}) = \int L_\Omega(y_i = j; \mathbf{v}; \boldsymbol{\omega}) g(\boldsymbol{\omega}|\boldsymbol{\eta}) d\boldsymbol{\omega} = \int e^{\mathbf{v}_j \boldsymbol{\omega}} / \sum_{j \in \Omega} e^{\mathbf{v}_j \boldsymbol{\omega}} g(\boldsymbol{\omega}|\boldsymbol{\eta}) d\boldsymbol{\omega}, \tag{8}$$

where $P_\Omega(y_i = j|\mathbf{v}, \boldsymbol{\eta})$, is a choice probability for alternative j in choice set Ω, and $\boldsymbol{\eta}$ is a vector of hyperparameters describing the probability density function of the mixing distribution $g(\cdot)$. $L_\Omega(j; \mathbf{v}; \boldsymbol{\omega})$ is a logit model for choice set Ω, $\boldsymbol{\omega} = \bar{\boldsymbol{\varpi}} + \boldsymbol{\Lambda}\boldsymbol{\zeta}$, denotes a $L \times 1$ vector of random coefficients, comprised of $\overline{\boldsymbol{\varpi}}$, a $L \times 1$ vector of *mean* parameters, Λ, a $L \times L$ matrix of second moment hyperparameters ($\boldsymbol{\Lambda}\boldsymbol{\Lambda}' = \Xi_\varpi$), and $\boldsymbol{\zeta}$, a $L \times 1$ random vector of *iid* variates with density $f(\boldsymbol{\zeta})$. The mixed logit log-likelihood for given $\boldsymbol{\omega}$ and $\boldsymbol{\eta}$ is then written as

$$l(\boldsymbol{\omega}, \boldsymbol{\eta}) = \sum_i \sum_j y_{ij} \log[\int_\omega \underbrace{\{e^{\mathbf{v}_{ij} \boldsymbol{\omega}} / \sum_{j \in \Omega} e^{\mathbf{v}_{ij} \boldsymbol{\omega}}\}}_{Q} g(\boldsymbol{\omega}|\boldsymbol{\eta}) d\boldsymbol{\omega}], \tag{9}$$

where $y_{ij} = 1$ (0) if individual i chooses (does not choose) alternative j. As an example, if the distribution over the parameters $\boldsymbol{\varpi}$ is driven by an independent normal mixing distribution then $\boldsymbol{\Lambda}$ is a diagonal matrix with a $L \times 1$ vector of standard deviations of the random coefficients along the diagonal.

Note that if we assume that $g(.|\boldsymbol{\eta})$ is multivariate normal then a mixed MNL model can be used to approximate a mixed probit model. We note that the advantage of MXL is that conditional upon integrating out the random taste heterogeneity, a tractable logit choice probability remains, as is evident by the term Q in (9).[7] The MNL model is delivered if we set to Λ to a null matrix. In this case the distributions over all coefficients are degenerate such that $\boldsymbol{\omega} = \bar{\boldsymbol{\varpi}}$ is a vector of fixed coefficients. In this regard we see that the MNL is the natural benchmark model with which to evaluate departures designed to incorporate different variants of the random coefficient model. The disadvantage of the MXL is that since the dimension of $\boldsymbol{\omega}$ is $L \times 1$, then if mixing is performed over all elements of $\boldsymbol{\omega}$, maximisation of the likelihood function involves the estimation of a $L-$dimension integral.

Note that when we explicitly allow for the panel structure the mixed logit log-likelihood for given $\boldsymbol{\omega}$ and $\boldsymbol{\eta}$ is written as

$$l(\boldsymbol{\alpha}, \boldsymbol{\eta}) = \sum_i \left[\sum_t \sum_j y_{ij}^t \log[\int_\omega \{e^{\mathbf{v}_{ij}^t \omega} / \sum_{j \in \Omega} e^{\mathbf{v}_{ij}^t \boldsymbol{\omega}}\} g(\boldsymbol{\omega}|\boldsymbol{\eta}) d\omega] \right], \tag{10}$$

where $y_{ij}^t = 1$ if individual i chooses the j^{th} alternative on the t^{th} choice occasion, where $t = 1, ..., T$. If we assume that preferences are constant for a given individual over the T choice occasions, then in simulating the probability for the i^{th} individual we make a draw from $g(\boldsymbol{\omega}|\boldsymbol{\eta})$, say $\boldsymbol{\omega}_{il}$, where l indexes the l^{th}attribute, and keep this fixed for all T.

Proposition 4 (Individual specific random components) *For the mixed logit model with J alternatives, and L characteristics with random taste variation over all L attributes, there are a maximum of $(L+1)L/2$ free error covariance parameters.*

The proof of this proposition follows from the same as Proposition 2. Note that in both cases the normalisation with respect to location does not impact upon the number of estimable parameters in Ξ_ω. Second, in both models the normalisation with respect to

[7] Note that for this reason the mixed multinomial logit (MXL) model, equivalently the random coefficient logit model, is sometimes referred to as Kernel Logit.

scale is automatic. For the mixed logit model this normalisation is automatic since the $\boldsymbol{\varepsilon}_i$ in (7) is i.i.d. Type 1 extreme value; and for the mixed probit model in question $\Xi_{\varepsilon J}$ has been set to an identity matrix, which encompasses both a scale plus additional overidentifying restrictions.[8]

3.5 Nested Logit

A number of generalisations of vanilla logit can be considered within a class of generalised extreme value (GEV) models. The most utilised departure is the nested logit (NL) model. We introduce the NL model by considering the decomposition of a joint distribution as a product of a marginal and a conditional distribution; and then link this decomposition to a hierarchical structure which overlays the original choice set. We begin by examining a simple two level hierarchy which divides the choice set into two partitions (or nests). Let the top level of this hierarchy define an aggregate choice over subset Ω_1 and Ω_2, where $\Omega_1 = \{a_1, a_2, \ldots, a_{J1}\}$, $\Omega_2 = \{b_1, b_2, \ldots, b_{J2}\}$, and $\Omega_1 \cup \Omega_2 = \Omega_J$. We write the indirect utility associated with choice $a_j \in \Omega_1$ as

$$U_{a_j} = V_{\Omega_1} + V_{a_j|\Omega_1} + \varepsilon_{\Omega_1} + \varepsilon_{a_j|\Omega_1}, \tag{11}$$

where both the deterministic and random components of utility are partitioned into two components: V_{Ω_A} and $V_{a_j|\Omega_1}$ represent, respectively, observed attributes of choice that depend on the top and lower level of the hierarchy; ε_{Ω_1} denotes the aggregate choice specific error component; and $\varepsilon_{a_j|\Omega_1}$ is the unobserved component which is specific to alternative a_j in the aggregate choice set Ω_1. Obviously the composite error term $\eta_{a_j} = \varepsilon_{\Omega_1} + \varepsilon_{a_j|\Omega_1}$ will, in analogous fashion to a standard error components formulation, generate non-zero covariance in unobserved utilities across all alternatives which belong to the same top level partition; covariance for alternatives $a_j \in \Omega_1$ and $b_j \in \Omega_2$ will be zero. This basic observation encapsulates the fundamental nature of the departure of NL from the vanilla logit model.

[8] See Eklof and Weeks (2003) for a number of important caveats.

To introduce the NL model formally we let $\boldsymbol{\eta} = (\boldsymbol{\eta}^{\Omega_1}, \boldsymbol{\eta}^{\Omega_2}, ..., \boldsymbol{\eta}^{\Omega_K})$ denote the vector of unobserved utility over partitions Ω_k, $k = 1, ..., K$, where $\boldsymbol{\eta}^{\Omega_k} = (\eta_{k_1}, \eta_{k_2}, ..., \eta_{k_J})$. The cumulative distribution of $\boldsymbol{\eta}$ may be written as

$$\exp - \left(\sum_{k=1}^{K} \left(\sum\nolimits_{j \in \Omega_k} e^{-\eta_j / \lambda_k} \right)^{\lambda_k} \right),$$

where k indexes the number of partitions and λ_k is a measure of the degree of independence in partition k. For any two alternatives in partition Ω_k we have a common factor ε_{Ω_k} in the stochastic component. Therefore, despite the fact that each element of η_j is distributed univariate extreme value, alternatives within the same partition are not independent. $\lambda_k = 1$ indicates independence of all alternatives in nest k. A test of the null hypothesis $\lambda_k = 1 \;\; \forall \; k$ represents a test of whether the NL model represents a statistically significant departure from vanilla logit. Different magnitudes of λ_k represent degrees of correlation among the unobserved portion of utility within each partition. As demonstrated inter alia by McFadden (1981) and Daly and Zachary (1981), the probability of choosing a_j in particular Ω_1 is given by

$$P_{a_j|\Omega_1} = \frac{e^{V_{a_j}/\lambda_{\Omega_1}} \left(\sum\nolimits_{j \in \Omega_1} e^{V_j/\lambda_1} \right)^{\lambda_1 - 1}}{\sum_{l=1}^{2} \left(\sum\nolimits_{j \in \Omega_1, \Omega_2} e^{V_j/\lambda_l} \right)^{\lambda_l}}. \tag{12}$$

For $a_j \in \Omega_1$ and $b_j \in \Omega_2$, the ratio of the respective probabilities is given by

$$\frac{P_{a_j|\Omega_1}}{P_{b_j|\Omega_2}} = \frac{e^{V_{a_j}/\lambda_1} \left(\sum\nolimits_{j \in \Omega_1} e^{V_{aj}/\lambda_1} \right)^{\lambda_1 - 1}}{e^{V_{b_j}/\lambda_2} \left(\sum\nolimits_{j \in \Omega_2} e^{V_{b_j}/\lambda_2} \right)^{\lambda_2 - 1}} \tag{13}$$

For alternatives a_j and a_l, $j, l \in \Omega_1$, the terms in parenthesis in (13) fall out, such that this ratio may be written

$$\frac{P_{a_j|\Omega_1}}{P_{a_l|\Omega_1}} = \frac{e^{V_{a_j}/\lambda_1}}{e^{V_{a_l}/\lambda_1}}, \tag{14}$$

which is independent of all other alternatives.

We note that this form of partitioning introduces a different form of the IIA assumption. Namely, for pairs of alternatives within the same partition the standard IIA condition applies, as in (14). However, across partitions the parenthetical terms in (13) are distinct. IIA is now relaxed in a particular way given that the ratio of probabilities, say $\frac{P_{a_j|\Omega_1}}{P_{b_j|\Omega_2}}$, depends on the attributes of a specific set of alternatives, namely those which share the same partitions as a_j and b_j. In this respect, as Train (2002) comments, we may modify the IIA statement, stating that an Independence of Irrelevant Nests (IIN) condition now holds over alternatives in different nests.

4 Ordinality and Discrete Response: Single Index Models

Multinomial probit and logit models are examples of *multiple index* models with *both* the unobserved (continuous) latent variable and observed discrete indicators indexed by the alternative. The assumption that there exists an ordering over Ω_J facilitates the specification of a *single* index ordinal response model.[9] In this instance the observational rule is characterized by the following mapping.

$$y_i = \kappa(y_i^*) = \mathbf{1}(\alpha_{j-1} < y_i^* = D_i + \varepsilon_i < \alpha_j) \cdot j \tag{15}$$

where y_i^* is a scalar quantity and α_j are thresholds. This gives rise to a single index, here D_i.

To motivate the use of ordinal response models, we consider the following example. Let $y_i^* = U_i(A) - U(B)$ denote the difference in value (or utility) for the i^{th} respondent from choosing product A over B. Using the mapping in (15) we write the observed discrete outcomes as: $y_i = 1$ (2) representing a choice *Definitely B* (*Probably B*); $y_i = 4$ (5) depicts the choice *Probably A* (*Definitely A*); and

[9] Note that for $J = 2$, the ordered response and binary models are equivalent.

$y_i = 3$ represents the region on $\Re$ where the respondent is indifferent between the two products.

In this instance the choice set represents an ordering which overlies the *scalar* utility difference y_i^*, in the sense that $y_i \in \{1, 2, ..., J\}$ represents as set of discrete ordinal indicators on a single unobserved latent variable y_i^*. We note that this example represents an observational rule which is consistent with the elicitation of *probabilistic* intentions. In particular, note that in a revealed preference setting stochastic uncertainty in the random utility model is predicated upon imperfect information on behalf of the analyst. However, in formulating the use of a discrete choice model to represent *stated* preferences, it is possible that consumers are also uncertain as to which alternative to choose given current information. (See Manski (1995) and Manski (1990)).

For a single respondent probabilities calculated using (15) are given by

$$\begin{aligned}
\Pr(y_i = j|\mathbf{x}_i) &= \Pr(\alpha_{j-1} \leq y_i^* < \alpha_j) \\
&= \Pr[((\alpha_j - \mathbf{x}_i'\boldsymbol{\delta})/\sigma) \leq \varepsilon_i \leq ((\alpha_{j-1} - \mathbf{x}_i'\boldsymbol{\delta})/\sigma)] \\
&= F((\alpha_j - \mathbf{x}_i'\boldsymbol{\delta})/\sigma) - F((\alpha_{j-1} - \mathbf{x}_i'\boldsymbol{\delta})/\sigma),
\end{aligned}$$

for $j = 1, ..., J$, and $\alpha_J = \infty$, $\alpha_0 = -\infty$. The log-likelihood is given by

$$\sum_{i=1}^{N} y_{ij}(\log[F((\alpha_j - \mathbf{x}_i'\boldsymbol{\delta})/\sigma) - F((\alpha_{j-1} - \mathbf{x}_i'\boldsymbol{\delta})/\sigma)].$$

where y_{ij} is 1 (0) for chosen (not chosen) alternatives. For $\varepsilon_i^A, \varepsilon_i^B$ distributed type 1 extreme value (normal) then $\varepsilon_i = \varepsilon_i^A - \varepsilon_i^B$ is also type extreme value (normal). DCM can estimate both ordered probit and ordered mixed probit models. Therefore, if we allow for $\boldsymbol{\delta}$ to be random (as opposed to fixed) coefficients, then the ordered mixed probit model is simply the ordered probit probabilities weighted by the density of δ, say $g(\boldsymbol{\delta}|\boldsymbol{\eta})$. In this case the log-likelihood is simply

$$\sum_{i=1}^{N} y_{ij} \log[\int [F((\alpha_j - \mathbf{x}_i'\boldsymbol{\delta})/\sigma) - F((\alpha_{j-1} - \mathbf{x}_i'\boldsymbol{\delta})/\sigma)]g(\boldsymbol{\delta}|\boldsymbol{\eta})d\boldsymbol{\delta}].$$

where, as discussed in section 3.4, $\boldsymbol{\eta}$ is a vector of hyperparameters describing the probability density function of the mixing distribution $g(\cdot)$.

4.1 Identification

As with the multinomial model, if the vector $\boldsymbol{\alpha}$ constitute parameters to estimate, the scale parameter σ is not separately identifiable. Subsequently, α_j/σ may be interpreted as a type of intercept for each probability, thereby precluding separate identification of an intercept in $\mathbf{x}_i^{'}\boldsymbol{\delta}$. It is obviously possible to separately identify a constant term in conjunction with $J-2$ threshold parameters if we introduce a normalisation such as setting one of the threshold values equal to zero. Therefore, in general there exists the following choice: either estimate $J-1$ composite thresholds of the form $(\widetilde{\alpha}_j = (\alpha_j - \gamma)$ OR set one threshold to zero and then estimate estimate $J-2$ thresholds plus γ. In DCM we impose the former identification condition.

5 Using the DCM class

This section demonstrates the use of the DCM class in Ox programs. The DCM class is derived from the `Modelbase` class and consequently inherits the main features of this class. However, there are some features specific to discrete choice models that do not fit the general `Modelbase` class and therefore some member functions of the `Modelbase` class have been modified to fit our purposes. In the following sections we will focus on model specification and estimation. In an accompanying manual we present the complete set of modified `Modelbase` member functions as well as the DCM member functions.[10]

[10] `http://www.econ.cam.ac.uk/faculty/weeks/DCM/DCMManual/`

5.1 Data organization

DCM can read any data format available in Ox and GiveWin. Furthermore, DCM accepts multiple types of data organizations, i.e., the organization of the data in the database. For example, in revealed preference or ordered models, data is usually organized such that each row in the data set refers to a specific individual-period observation where the columns hold both individual characteristics and possible alternative attributes. Individual characteristics can be either in a single column or repeated in J columns. The dependent variable is given either in a single column as the index of the chosen alternative, or in J columns with a non-zero value in the column corresponding to the chosen alternative. On the other hand, in stated preference models, the organization is often such that each row refers to a specific individual-period-alternative observation and the dependent variable is given in a single column indicating with a non-zero value the chosen alternative. Observations are finally stacked by alternative, by time periods, and by individuals.

Most software programs can handle one of the possible combinations. With DCM, this has all been taken care of. From the information on the dependent variable, DCM infers the organization of the data without any input from the user. The data is then internally transformed to a unified framework. Hence, DCM can interpret most permutations of data organizations: rows can refer to individual-period or individual-period-alternative observations,; the dependent variable can either an index or an dummy variable (the index can start at any number); and individual characteristics can be in a single column or repeated across columns. The cost of this property is that DCM needs variable names for all columns in the dataset. Columns that refer to the same variable must have the same names such that names are repeated across columns. Also, given the number of time periods in a balanced panel, DCM will infer the number of individuals and the number of alternatives. In the case of unbalanced panels users can create an individual ID variable which is indicated to DCM using `SetID(sPar)`. DCM is then able to infer the number of individuals, time periods and alternatives from this variable.

In essence, the requirements on data organization in DCM are as follows: *a*) observations should be stacked by period and by individuals; *b*) each column should have a name that reflects the organization of the data[11]. In unbalanced panels, there should be one variable indicating the ID of the individual.

5.2 Model specification

Once the data has been loaded and variables selected,[12] the user must make a number of choices over type of model and specification of the stochastic components. Below we discuss the following DCM member functions:

`SetModel(iModel)`, `ScaleVar(aScaleVar)`,
`SetRefAlt(iAlt)`, `SetScaleAlt(iAlt)`,
`SetCoeffDist(iType,asPar,bCorr)`, and
`SetErrDist(iType)`.

We first begin with the `SetModel()` statement and the options available for each model

SetModel(iModel) Selects the model to be estimated. The available options for `iModel` are presented in Table 1 with the total number of model parameters[13]. Note that all model parameters are not identified. Table 2 gives an overview of the options available for the various models. These options are discussed in detail below.

SetRefAlt(iRefAlt) Sets the reference alternative, i.e., the alternative against which other alternatives' utilities are compare. This necessary for identification with respect to location. Note

[11]There is one caveat in DCM 1.0. In the OxPack implementation we are forced to restrict the available data organizations to the case where observations are stacked by alternatives, by periods, and by individuals (c.f. stated preference data). However, there is a member function in the DCM class that transforms any dataset to this organization. For details, we refer the reader to the DCM manual.

[12]See the documentation on the `Modelbase` class for loading and selecting variables.

[13] F (R) denotes the number of fixed (random) coefficients.

Table 1: SetModel() options

iModel	Description	#par
M_CL	Conditional/multinomial logit	$(J-1)K+L$
M_NL	Nested logit	$(J-1)K+L+\#nests$
M_MXL	Mixed logit	$F+R+R(R+1)/2$
M_MXP	Mixed probit	$F+R+R(R+1)/2$
M_MNP	Multinomial Probit	$F+R+R(R+1)/2+J(J-1)/2-1$
M_OP	Ordered probit	$F+J-1$
M_OMP	Ordered mixed probit	$F+R(R+1)/R+J-1$

Table 2: Model specifications

SetModel(iModel)		CL	NL	MXL	MXP	MNP	OP	OMP
SetRefAlt(iRefAlt)		✓	✓	✓	✓	✓	n.a.	n.a.
(D:	iRefAlt=0)							
SetScaleAlt(iScaleAlt)		n.a.	n.a.	n.a.	✓	✓	n.a.	n.a.
(D:	iScaleAlt=1)							
SetNest(avNest)		n.a.	✓	n.a.	n.a.	n.a.	n.a.	n.a.
SetCoeffDist(iType,asPar,bCorr)								
iType =	FIXED	(A)	(A)	✓(D)	✓(D)	✓(D)	(A)	(A)
	NORMAL	n.a.	n.a.	✓	✓	✓	n.a.	✓
	LOGNORMAL	n.a.	n.a.	✓	n.a.	n.a.	n.a.	✓
SetErrDist(iType)								
iType =	ED_IID	n.a.	n.a.	(A)	(A)	✓(D)	(A)	(A)
	ED_HETEROSC	n.a.	n.a.	n.a.	n.a.	✓	n.a.	n.a.
	ED_UNREST	n.a.	n.a.	n.a.	n.a.	✓	n.a.	n.a.
SetRandom(iType,cR)								
iType =	R_UNIFORM	n.a.	n.a.	✓	✓	✓	n.a.	✓
	R_HALTON	n.a.	n.a.	✓(D)	✓(D)	✓(D)	n.a.	✓(D)
	R_NONE	n.a.	n.a.	n.a.	✓	✓	n.a.	n.a.
(D:	cR=50)							

Note:✓=Option available, n.a.=Option not available, A=Automatic, D=Default that Ox indices starts at zero, hence the first alternative will have index 0. The variance of the error term for the `iRefAlt` is set to 1 in the MNP model. `iRefAlt` cannot be the same as `iScaleAlt` (see below).

SetScaleAlt(iScaleAlt) Sets the scale alternative of the model i.e. the alternative for which the diagonal element in the cholesky decomposition of the error covariance matrix is set to unity. This is only relevant for models with normally distributed error terms. This cannot be the same alternative as `iRefAlt` (see above).

SetNest(avNest) Sets the nesting structure of the model.[14] The argument is an array of vectors with alternative indexes. The current version of DCM only supports one level of nesting, i.e. nests within nests are not supported.

SetCoeffDist(iType,asPar,bCorr) Sets the mixing distribution of random coefficients. The first argument defines the distribution of the parameters listed in the second argument. The third optional argument allows for correlated random coefficients. This function can be called several times to specify different distributions for different sets of coefficients.

As the log-normal distribution is defined with a *positive support*, in order to model e.g. a *negative* price effect, it is necessary to create a price variable that has a negative support.

SetErrDist(iType) Sets the structure of the error covariance matrix. All available options in `DCM` will result in an identified error covariance matrix. See Appendix B.1 for a detailed description of the error covariance structure.

SetRandom(iType,cR) Sets the type and number of random draws for models that requires Monte Carlo integration techniques (MXL, MXP, MNP, and OMP). For multiple index models

[14] Only relevant for `cModel=M_NL`.

where the dimension of Ω_J is three or less, and assuming normally distributed errors, the user can choose the option `R_NONE` and thereby utilise the Ox numerical integration routines for the univariate and multivariate normal probabilities.

5.2.1 Comments

One of the principle distinctions between the mixed logit and multinomial probit model can be appreciated from Table 2. For example, consider the case where an analyst is contemplating estimating either a mixed logit or multinomial probit model, and is interested in allowing for random preference heterogeneity. In the case of the `M_MXL` model the user will use `SetCoeffDist` to make choices as to whether individual mean coefficients are to be considered as fixed or random; and, if random, both the form of the mixing distribution and whether random components are correlated, needs to be specified. However, the use of the `M_MNP` model requires both this setting in conjunction with a specification on the *residual* error component using `SetErrDist`. This follows from the fact that whereas the `M_MXL` model partitions the stochastic component into two additive parts - one heteroscedastic and correlated over the choice set, and another which is i.i.d. type 1 extreme value, the `M_MNP` model does not make such a distinction. Two other observations are worth making. First, one disadvantage of the MNP model is that the form of the mixing distribution is fixed and normal. In the current version of the paper, a mixed logit model comes with two mixing distributions: normal and log-normal. Second, if a MNP model is chosen *and* a user allows for free covariance parameters, identification restrictions are required. In contrast the mixed logit model has at its core a kernel logit model, and therefore identification is automatic.

5.3 Estimation

This subsection discusses the following member functions:
`SetAlgorithm(iAlg),SetStdErr(iType),`

Table 3: SetAlgorithm() options

iAlg	Description
A_BHHH	BHHH optimizer that comes with DCM (default)
A_BFGS	Ox's intrinsic MaxBFGS optimizer
A_NEWTON	Ox's intrinsic MaxNewton optimizer
A_SQP	Ox's intrinsic MaxSQP optimizer

Table 4: SetStdErr() options

iType	Description
SE_ROBUST	Robust standard errors used (default)
SE_HESSIAN	Standard errors calculated using the Hessian
SE_OPG	Standard errors calculated using the outer product of the gradient

`SetStartPar(vP)`, and `Estimate()`.

SetAlgorithm(iAlg) Sets the optimization algorithm. Table 3 presents the available options.

SetStdErr(iType) Sets the standard errors. Table 4 presents the available options.

SetStartPar(vP) Sets the vector of starting values. Note that `DCM` by default constructs starting values using either CL (for `multiple` index models requiring simulation) or OP (for `single` index models requiring simulation). This is overridden by `SetStartPar()`. The organization of the parameter vector is described in Table 5.

Estimate() Estimates the specified model.

Examples of these commands will be given in the following section on examples.

5.4 Post-estimation analysis

In DCM there is a single post-estimation option for analysis.

TestRandCoeff(const asPar) Tests for individual heterogeneity in the coefficients listed in the the array `asPar`. If `asPar=-1`,

Table 5: Organization of vector of starting values

# of pars.	Description
$(J-1)$	alternative specific constants (`Deterministic(TRUE)`)
$(J-1)\cdot K$	individual characteristics ordered as in the `Select(I_VAR,...)` statement
L	attributes ordered as in the `Select(A_VAR,...)` statement
M	interaction terms ordered as `asV[1]`·`asW[1]`, `asV[1]`·`asW[2]`, ...,`asV[k]`·`asW[l]`
$\#nest$	inclusive values (NL)
$J-1$	thresholds (OP and OMP)
Q	estimated elements of stacked columns of lower triangular cholesky decomposition of random coefficients covariance matrix (MXL, MXP, and MNP)
P	estimated elements of stacked columns of lower triangular cholesky decomposition of normalized error covariance matrix (MNP).

all coefficients are tested for heterogeneity. `asPar` can also be a vector of coefficient indices.

The use of these commands will be demonstrated in the following section on examples.

6 Examples

In the following section we present some examples which demonstrate the use of DCM in estimating parameters of a number of discrete discrete choice models. The focus of the presentation is not the modelling and interpretation of the estimates, but the syntax of the DCM commands the output produced by DCM.

6.1 Multinomial Models of Labour Force Status

To demonstrate the specification and estimation of multinomial probit models we use the trinomial discrete choice model of the labour force status of married women in the UK as considered by Duncan and Weeks (1998). The model is discrete in that they allow for three states: non-workers supplying zero hours of work; part-time workers whose weekly supply is between 0 and 30 hours; and full-time workers supplying more than 30 hours. The data consist of a random sample of married women drawn from the 1993 Family Expenditure Survey (FES).

Across all specifications we condition our labour supply model on wage rates[15], and the following socio-demographic *characteristics*; age of the woman, dummies for children in the age groups 0-2, 3-4, 5-10, and above 11, number of children, level of formal education and marital status (whether married or cohabiting). We also include a single *attribute* variable, net incomes at various hours levels. To generate state-specific net incomes as condition variables for the structural discrete choice models, we simulate tax liabilities and benefit receipts and total net incomes at 0, 20 and 40 hours for each individual in our sample.[16] Finally, to allow for age dependent income effect we interact net income and age. Our dependent variable is a three-state variable which distinguishes non-participants (category 0), part-time workers between 1 and 30 hours (category 1) and full-timers working in excess of 30 hours (category 2). The reference alternative is the non-participation category. In reading the economic significance of the parameter estimates, a negative coefficient represents a decrease in the likelihood of working either part-time or full-time relative to not working. Which comparison is appropriate is identified for each parameter estimate in the table. The DCM commands for estimating the model are

[15]Since wage rates are not observed in the FES for those not in employment, we base our simulations on wage rate estimates derived from an appropriately corrected reduced form equation. See Duncan and Weeks for further details.

[16]The wage equation is identified from the inclusion of demand-side (quarterly unemployment) and regional characteristics (vacancies and redundancies by region) as well as socio-demographic characteristics (quadratics and interactions between age, partners's age and education; age and number of children). Estimates are available from the authors on request. A problem with this approach is that it becomes difficult to correct the standard errors in the structural model for the inclusion of the generated wage rate term, since the simulated net income terms also depend (non-linearly) on the wage rate used. In the structural models, therefore, the standard errors remain uncorrected.

presented in Table 6.[17]

Table 6: A discrete choice model of labor supply: Multinomial Probit

```
1  #include "/packages/dcm/dcm.ox"
2  main()
3  {
4      fopen("LS_MNP.OUT","l");
5      decl obj = new DCM();
6      obj.Load("../data/Married4.xls");
7      obj.Select(Y_VAR,{"REFPRED"  ,0,0}); // Y_VAR: Dependent variable
8      obj.Select(I_VAR,{"DKID02"   ,0,0,   // I_VAR: Individual characteristics
9                        "DKID34"   ,0,0,   // Automatically interacted with an
10                       "DKID510"  ,0,0,   // alternative specific constant.
11                       "DKID110"  ,0,0,
12                       "TOT_KIDS" ,0,0,
13                       "AGE"      ,0,0,
14                       "EDGT16"   ,0,0,
15                       "COHAB"    ,0,0,
16                       "LNWFIT"   ,0,0});
17     obj.Select(A_VAR,{"INC",0,0});       // Alternative attributes
18     obj.Interact({"AGE"},{"INC"});
19     obj.ScaleVar({"INC",0.01});
20     obj.SetAlgorithm(A_BFGS);
21     obj.SetModel(M_MNP);
22     obj.SetErrDist(ED_UNREST);
23     obj.Estimate();
24     delete obj;
25 }
```

In line 1 we include the DCM class in the program.[18] Line 4 opens a log-file to which the output will be printed in addition to the screen. Line 5 creates the object `obj` in which we will load the data set (line 6) and select the variables (line 7-17). There are three groups of variables in DCM reflecting the canonical model presented in previous sections; the dependent variable (`Y_VAR`), individual characteristics (`I_VAR`), and alternative attributes (`A_VAR`). A variable in the loaded data set is selected in to one of these groups using the `Select(iGroup,...)` command.[19] The dependent

[17]See Duncan and Weeks (1998) for a discussion of parameter estimates and the application of both nested and non-nested tests across models.

[18]The DCM code includes `oxstd.h` and `oxfloat.h` automatically so the user does not need to include these header files as well.

[19]The arguments in the `Select()` statements use the `Modelbase` class syntax. Hence,

variable ($REFPRED$) is selected into the `Y_VAR` group, whereas the individual characteristics ($DKID02$, etc) and the alternative attribute (INC) are selected into the `I_VAR` and `A_VAR` groups, respectively. Further, line 18 generates and includes the interaction term between AGE and INC. The variable will be named `AGE*INC` in the printout.[20] Finally, the `ScaleVar({''INC'',0.01})` statement will multiply the INC variable by a factor 0.01 and we will use the BFGS algorithm as stated in line 20. [21]

The output from DCM presented in Table 7 includes information on the sample used for estimation and how DCM has interpreted the data organization.[22] The type and number of pseudo-random numbers in use are indicated, as well as the relevant base and scale alternatives. In the table of estimates, the label `C(Err)[i,j]` refers to the (i, j)'th element of the cholesky decomposition of the error covariance matrix as indicated by the note below the table. Below the tables of estimates and likelihood values, DCM reports the results from the initial estimation of a restricted conditional logit model.

We may also consider a number of alternative model specifications by focusing upon the stochastic component of choice. For example a mixed probit model (MXP) allows for heterogenous preferences but assumes that the error covariance matrix is iid. Here we allow for heterogenous preference over the income variable. In the DCM code, this is implemented by replacing `SetErrDist(ED_UNREST)` by `SetErrDist(ED_IID)` in line 22 and specifying the distribution of the relevant coefficients, e.g., `SetParDist(NORMAL,{''INC''});` before the `Estimate()` statement.[23]

It is also possible to estimate a mixed logit model (MXL) which accommodates heterogeneous preferences but assumes that the ad-

---

the second and third arguments (0,0) are not used in DCM.

[20]If any of the two arguments is an array, then all possible combinations of interaction terms will be created and included in the model.

[21]If several variables should be scaled, multiple calls to `ScaleVar()` can be made, or the user can send an array in a single call, e.g. `ScaleVar({''INC,0.01,''AGE'',100});`.

[22]If the data is a unbalanced panel, then information on the number of observations per cross section unit will be reported.

[23]We could also replace `SetModel(M_MNP)` by `SetModel(M_MXP)`, which is equivalent to `SetModel(M_MNP)` and `SetErrDist(ED_IID)`.

Table 7: Fragment of output from DCM estimation.

```
Ox version 3.30 (Windows) (C) J.A. Doornik, 1994-2003
DCM package version 040109, object created on 20-01-2004
NOTE: The number of alternatives is low (<3). For speed and
      efficiency, DCM recommends numerical integration
      Use SetRandom(R_NONE,0)
DCM package version 040109, object created on 20-01-2004
** NOTE: Estimating a Conditional Logit model to generate starting values...Done! **

---- DCM: Multinomial Probit ----
The estimation sample is:  1 - 1520
The dependent variable is: REFPRED  (../data/Married4.xls)
Data type : NT x (J or 1)K+JL+(J or 1) with
    N       = 1520
    J       =    3
Pseudo random draws:   HALTON (R=50)
Base alternative:         0
Scale alternative:        1

                  Coefficient  Std.Error  t-value  t-prob
(0)  DKID02 1/0      -2.84209      1.886    -1.51   0.132
(1)  DKID02 2/0      -19.5861      6.027    -3.25   0.001
(2)  DKID34 1/0       12.5196      2.167     5.78   0.000
...
(18) INC              3.56419     0.8131     4.38   0.000
(19) AGE*INC     -0.000311216   0.003139  -0.0991   0.921
C(Err)[0,0]           1.00000    (fixed)
C(Err)[1,1]           1.00000    (fixed)
C(Err)[1,2]          0.362283     0.5711    0.634   0.526
C(Err)[2,2]           1.47302     0.5381     2.74   0.006

NOTE: Robust standard errors.
NOTE: C[i,j] denotes entries in the Cholesky decomposition of the covariance matrix.
NOTE: Following variables are scaled.
    INC x 0.01

Error term:
            Std.dev Correlations
ALT_0        1.0000       1.0000      0.00000      0.00000
ALT_1        1.0000      0.00000       1.0000      0.23883
ALT_2        1.5169      0.00000      0.23883       1.0000

log-likelihood    -130.418434
no. of observations      1520  no. of parameters          22
AIC.T             304.836869  AIC              0.200550572

Starting values generated from CL estimates: Strong convergence, lnLoglik =-129.19
Time to convergence:    5:26.79   (hh:mm:ss.hs, excluding time for covariance.)
BFGS using numerical derivatives (eps1=0.0001; eps2=0.005):
Strong convergence
Used starting values:
      -3.2131      -18.396       11.385      -21.563       26.807      0.23972
       22.805       3.6487      -5.9604      -5.6486       2.2276      -16.308
    -0.054199    -0.010512      -7.6255       3.7605       6.3609       15.216
       3.3098  -0.00038376      0.00000       1.0000
```

ditive (Type 1 Extreme Value) disturbance term is iid across both alternatives and individuals. In the DCM code, this would be accomplished by replacing `SetModel(M_MNP)` by `SetModel(M_MXL)` in line 21 and specifying the coefficient distribution, e.g. `SetCoeffDist(NORMAL,{''INC''})`. The `SetErrDist(ED_UNREST)` statement can be dropped.

The user may also append the code for the additional models to the original program, where only changes across models is required. A code fragment which replace lines 21-23 in the previous program is given in Table 8. Note that we need to restate settings that are model specific, e.g. the distribution of the random coefficient. This is related to the fact that not all mixing distributions are allowed in all types of models. We have also used numerical integration in the MXP model, and 100 Halton draws in the Monte Carlo integration in the MXL model.[24]

Table 8: Multiple specifications in one file, code fragment

```
//  This fragment replaces line 21-23 in Table 6.
//  Unrestricted multinomial probit model
    obj.SetModel(M_MNP);
    obj.SetErrDist(ED_UNREST);
    obj.Estimate();
//  Mixed probit model with mixing over income
    obj.SetModel(M_MXP);
    obj.SetRandom(R_NONE,0);
    obj.SetCoeffDist(NORMAL,{"INC"});
    obj.Estimate();
//  Mixed logit model with mixing over income
    obj.SetModel(M_MXL);
    obj.SetRandom(R_HALTON,100);
    obj.SetCoeffDist(NORMAL,{"INC"});
    obj.Estimate();
```

6.2 Choice of Transport Mode

Following earlier work by Daganzo (1979) and McFadden (1977), the transport mode choice problem has continued to figure prominently

[24]The output is not reported here but can be requested from the authors.

in both reflecting and promoting developments in discrete choice methodology. In recognition of this fact, we now examine a well known modal choice dataset recording the inter-city travel choices between Melbourne, Canberra and Sydney. There are a total of 210 observations of non-business travellers faced with the choice between plane, car, bus, and train.[25] This dataset has been used extensively with examples including Greene (2002), Louviere, Hensher, and Swait (2000), and Ben-Akiva, Bolduc, and Walker (2001). The covariates included are terminal waiting time ($Ttme$), in-vehicle cost ($Invc$), in-vehicle time ($Invt$), generalized costs (GC) calculated from $Invt$, $Invc$, and a measure of wage rates, and household income ($Hinc$) interacted with the "air" alternative. The estimated model is thus

$$U_j = \alpha_j + \beta_1 GC_j + \beta_2 Ttme_j + \beta_3 (Hinc * air_j) + \epsilon_j$$

where α_j is an alternative specific constant and air_j is a alternative specific dummy.

We estimate four models. The first model is a simple conditional logit model, followed by a test for random preference heterogeneity. The code for this model is presented in Table 9.

Here we have introduced a number of new DCM commands. The `SetAltNames()` command is used to set the names of the alternatives. These names are used to refer to the various alternatives and for print-outs. The `Deterministic(TRUE)` command is used to generate and include alternative specific constants in the regression. The constants can be used to create interaction terms using `Interact()` commands (see above). If no alternative names are given, DCM will create dummies named e.g. `ALT_0`. The `TestRandCoeff(-1)` command will test all coefficients for heterogeneity.[26] The test in performed as a F-test for excluding the arti-

[25]Note that the dataset is actually choice-based, with undersampling of the more popular mode, car. In order to obtain consistent estimates a weighted exogenous sample maximum likelihood estimator (WESML) should be used. However, we do not do this, since our primary objective is to compare our estimates with those in Greene (2002) and Louviere, Hensher, and Swait (2000).

[26]To test a subset of coefficients use e.g. `TestRandCoeff({''GC'',''Ttme''})`.

Table 9: Travel mode example. Conditional logit estimates and test for heterogeneity

```
1  #include "/packages/dcm/dcm.ox"
2  main()
3  {
4      decl obj = new DCM();
5          obj.Load("../data/Greene_0.xls");
6          obj.SetAltNames({"air","train","bus","car"});
7          obj.Deterministic(TRUE);
8          obj.SetRefAlt(3);  /* Reference alt. is "car" */
9          obj.Select(Y_VAR,{"Mode",0,0});
10         obj.Select(A_VAR,{"GC"  ,0,0,
11                           "Ttme",0,0});
12         obj.Interact({"Hinc"},{"air"});
13         obj.ScaleVar({"GC",0.1});
14         obj.SetAlgorithm(A_BFGS);
15         obj.SetModel(M_CL);
16         obj.Estimate();
17         obj.TestRandCoeff(-1);
18     delete obj;
19 }
```

ficial variables.[27]

We follow this by estimating a nested logit and two mixed logit models. The first applies independent normal mixing distributions to each variate, and we then follow this by allowing random parameters to be correlated. The additional required DCM commands are presented in Table 10 which replace line 15-17 in Table 9.

6.3 DCM in OxPack for GiveWin

In this section we provide an overview of the graphical user interface of DCM using OxPack for GiveWin. The user need a license to run OxPack via GiveWin. This section presents the estimation of a MXL model using the transportation mode example discussed in the previous section. For a detailed description of all available options and dialogs in DCM in OxPack we refer the reader to the DCM manual.

[27]The output of the program is not presented here but can be requested from the authors.

Table 10: Travel mode example, cont.

```
//      Conditional logit model
        obj.SetModel(M_CL);
        obj.Estimate();
        obj.TestRandCoeff(-1);
//      Nested logit
        obj.SetModel(M_NL);
        obj.SetNest({<0>,<1,2,3>});
        obj.Estimate();
//      Mixed logit model, independent coefficients
        obj.SetModel(M_MXL);
        obj.SetCoeffDist(NORMAL,{"air","train","bus"});
        obj.Estimate();
//      Mixed logit model, correlated coefficients
        obj.SetModel(M_MXL);
        obj.SetCoeffDist(NORMAL,{"air","train","bus"},TRUE);
        obj.Estimate();
```

We assume that the database has been loaded into GiveWin and that all the necessary variables have been created. Note that, in contrast to the stand-alone version of DCM, the OxPack implementation of DCM is restricted w.r.t. the data organization. The OxPack version can only handle the $NTJ \times K + L + 1$ organization. We will also assume that DCM has been loaded into OxPack using `Package-Add/Remove packages` in OxPack, then selected using `Package-DCM`.[28] If DCM is successfully loaded, the OxPack dialog box will indicate `DCM` in the title bar and there will be a message in the GiveWin result window.

DCM in OxPack follows the conventional sequence of commands in OxPack. That is 1) model selection, 2) model formulation, 3) model settings, 4) estimation, and 5) post-estimation testing.

1. **Model selection** First we select the model to estimate. This is done from the "Model" menu as illustrated below. Once selected, the user will be taken automatically to the `Data selection...` dialog.

2. **Formulate/Data selection** Next, as illustrated in Figure 2,

[28]Use `Add/Remove package...` in the OxPack menu `Package`. Then locate and add DCM.OX to add the DCM package.

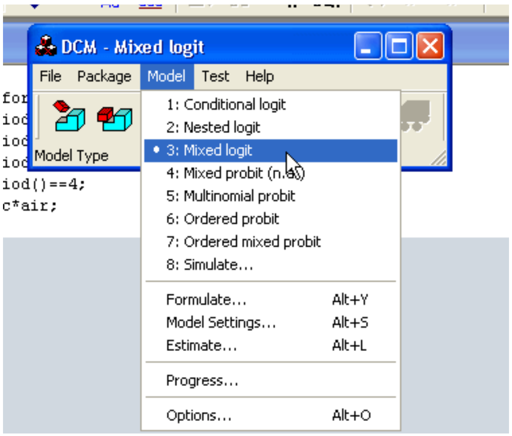

Figure 1: Model selection dialog

we select the dependent and independent variables from the data set loaded in GiveWin. Depending on the model, the left panel gives different options for the variables. In the MXL case, there are 7 available groups. The first three, `Y`, `A`, and `I` refer to the dependent variable, alternative attributes, and individual characteristics, respectively. As each variable is selected, an indicator will show up to the left of the variable[29].

The next three groups refer to the distribution of the random coefficient associated with the corresponding variable. For MXL models, there are three alternatives, fixed (non-random, default), normal and log-normal distribution. Selecting a variable in the model and clicking the desired distribution defines the mixing distribution. In the example we have set the distribution of the alternative specific constants to normal. The `SetID` function is used in unbalanced panel to indicate an individual id variable. Clicking `OK` takes the user to the `Model Settings` dialog.

3. **Model settings** The options in the `Model settings` dialog are different across models. For example, in the MXL model the user can change the type of covariance estimator, set

[29] As the first variable is selected into the model, DCM will automatically include alternative specific constants, which can be removed from the model using `Delete` if marked.

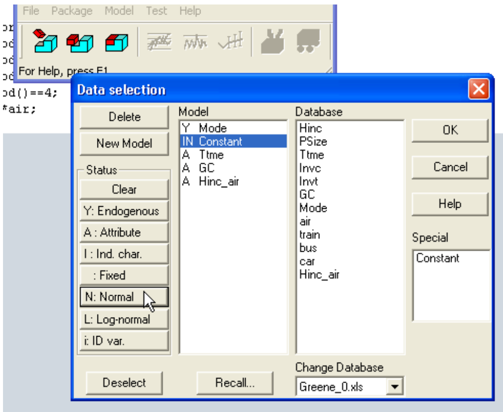

Figure 2: Formulation/Data selection dialog

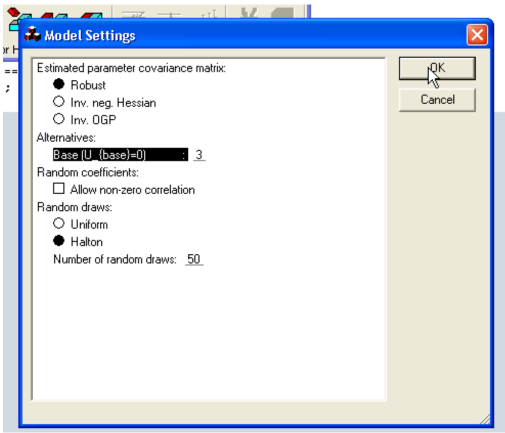

Figure 3: Model settings dialog

the reference alternative (note that indexing starts at 0, not 1), allow for correlation across random coefficients,[30] and the type and number of random draws used in the Monte Carlo integration. In this example, (see Figure 3) we use a robust covariance estimator, the `air` alternative (alternative 4, indexed as 3) is the reference alternative, we do not allow for correlation across random coefficients, and we use 50 Halton draws in simulations.

[30]Note that a current limitation of the oxPack implementation of DCM, is that unlike the standard version of DCM, where individuals write small programs in Ox (as demonstrated above), it is not possible to allow for correlation across subsets of random coefficients.

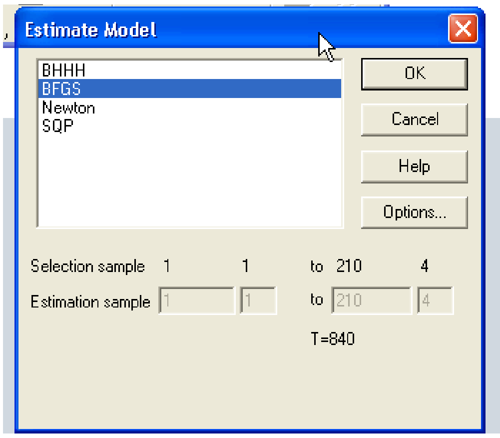

Figure 4: Estimation dialog

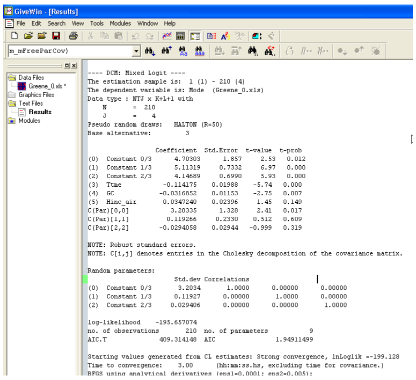

Figure 5: Output in GiveWin

4. **Estimation** The final step is to set the type of optimizer for estimation. There are four options as indicated in the figure below. Further, the user can click `Options...` to control the printout during estimation. Clicking `OK` will estimate the model and print the results to GiveWin's result window as illustrated in the Figure 4.

5. **Post-estimation testing** The user can test the usual exclusion and linear restriction test from the `Test` menu.

The user can now go back to the `Formulation` dialog to remove or add new variables and re-estimate the model.

7 A Word of Warning

With the continued increased in computer resources available for economic research, the potential of simulation technology to expand the set of viable models of choice behaviour has been realised. Access to software which includes the mixed multinomial logit and multinomial probit is now widespread with computer packages such as the `NLOGIT, HLOGIT` and now `DCM`. However it is important to offer a word of warning as to the likely impact of an expanding model choice set on research and ultimately policy decisions. As an example, Hensher and Greene (2002) note ten key specification issues that must be considered prior to estimating a random coefficient logit model. Three key aspects are: selecting the parameters that are to be random (fixed); selecting the distribution of the random parameters; and accounting for correlation over multiple choices made by an individual. Therefore, as the analyst seeks to accommodate the various manifestations of uncertainty in the random utility model, it is also appropriate to recognise that the analyst will have little theoretical guidance as to the most appropriate choice over these types of specification issues. Although obvious exceptions to this statement exist, for example avoiding distributions with positive support in representing consumer heterogeneity over product price, in many cases prior information is absent. As experience in the use of these models in both stated and revealed preference environment accumulates, this source of specification uncertainty will obviously fall.

In the preceding discussion we have noted the problem of *theoretical* identification in terms of locating which parameters are identified in the population model. However, in confronting any model with data, especially those with a relatively large number of covariance parameters, the analyst may be faced with problems of *empirical* identification. Over a wide class of discrete choice models, but with particular emphasis on the MNP model, this problem has been noted by a number of authors including McFadden and Train (2000). Keane (1992) refers to this problem as *fragile* identification. The key issue here is that after the necessary set of scale and level restrictions have been imposed, the analyst may experience problems in precisely estimating covariance parameters. One manifestation of this problem is that a number of competing model specifications, for example, those based on different covariance specifications, may deliver models with *almost* identical measures of fit, including the log-likelihood value.

References

BEN-AKIVA, M., D. BOLDUC, AND J. WALKER (2001): "Specification, Identification and Estimation of the Logit Kernel (or Continuous Mixed Logit) Model," Working Paper, MIT. http://web.mit.edu/jwalker/www/home.htm. Accessed April 1, 2002.

BHAT, C. (2003): "Simulation Estimation of Mixed Discrete-Choice Models Using Randomized and Scrambled Halton Sequences," *Transportation Research*.

BHAT, C. R. (1999): "Quasi-Random Maximum Simulated Likelihood Estimationof the Mixed Multinomial Logit Model," *Transportation Research*.

BUNCH, D., AND R. KITAMURA (1989): "Multinomial probit estimation revisited: Testing of new algorithms and evaluation of alternative model specifications for trinomial models of household car ownership.," Discussion Paper Research Report UCD-TRG-RR-4., Transportation Research Group, University of California Davis CA.

DAGANZO, C. (1979): *Multinomial Probit: The Theory and Its Application to Demand Forecasting*. Academic Press.

DUNCAN, A., AND M. WEEKS (1998): "Non-Nested Models of Labour Supply with Discrete Choices," Working Paper, Department of Economics, University of York .

EKLOF, M., AND M. WEEKS (2003): "On Specification, Identification and Estimation in a Class of Discrete Choice Models," Cambridge Working Papers in Economics, No. 0352.

GREENE, W. H. (2002): *Econometric Analysis Fifth Edition*. Prentice Hall, Upper Saddle River, New Jersey.

HENSHER, D. A., AND W. H. GREENE (2002): "The Mixed Logit Model: The State of Practice," ITS Working Paper 02-01. The

Institute of Transport Studies (C37) Faculty of Economics and Business The University of Sydney.

Louviere, J. J., D. A. Hensher, and J. D. Swait (2000): *Stated Choice Methods: Analysis and Application*. Cambridge University Press.

Manski, C. (1990): "The Use of Intentions Data to Predict Behavior: A Best Case Analysis," *Journal of the American Statistical Association*, 85, 934–940.

Manski, C. F. (1995): *Identification Problems in the Social Sciences*. Harvard University Press, Cambridge, Massachusetts.

McFadden, D. (1989): "A Method of Simulated Moments for Estimation of Discrete Response Models Without Numerical Integration," *Econometrica*, 57, 995–1026.

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

Pakes, A., and D. Pollard (1989): "Simulation and Asymptotics of Optimization Estimators," *Econometrica*, 54, 755–785.

Train, K. (1999): "Halton Sequences for Mixed Logit," mimeo, Department of Economics, University of California, Berkely.

——— (2002): *Discrete Choice Models with Simulation*. Cambridge University Press.

A Loading and modifying data

This subsection discusses the loading and modification of data in DCM. We make the distinction between the concepts *data format* and *data organization. Data format* refers to the file type in which the database is saved, e.g. an Excel spreadsheet or a Stata file. *Data organization* refers to how the data is organized *within* the database.

Data formats: The DCM class is derived from the `Modelbase` and `Database` class, and as a result, DCM can read all data formats that the `Database` class can handle. These data formats include Excel, Lotus, formatted and unformatted ASCII files, Stata, and GiveWin formats. We refer to the Ox manual for a description of available data formats.

Data organisation: There exist a number of different conventions to organize data in discrete choice models. We accommodate most conventions in `DCM`. Assuming that the dataset contains information on N individuals, T time periods, and J alternatives; and that we have K individual characteristics, and/or L alternative attributes, and a single dependent variable, then we identify two main types of data organization:

Type 1 $(NTJ \times K + L + 1)$ This organization is common in stated preference analysis of `multiple index` models. Observations are stacked by alternatives, by time periods, and by individuals. Each row includes K columns for individual characteristics, L columns for attributes and 1 column indicating the alternative that is preferred. The same ordering of alternatives in rows is applied for all individuals and time periods. See Greene1.xls for an example.

Type 2 $(NT \times (J \textbf{ or } 1) \cdot K + J \cdot L + (J \textbf{ or } 1))$ This organization is common in revealed preference analysis and `single index` models. Observations are stacked by time periods, and by individuals. Each row includes $J \cdot K$ or K columns for individual characteristics, and/or $J \cdot L$ columns for alternative attributes, and J or 1 column(s) indicating the preferred alternative. The same ordering of alternatives in columns applies to all individuals and time periods. If data for the dependent variable is supplied as a single column, then the dependent variable is assumed to represent the index of the chosen alternative, i.e. 1,2,3,...,J.[31] If the dependent variable spans multiple columns it is assumed to be a (row) vector with a non-zero value in the position of the preferred alternative. See Greene2.xls for an example.

Remark 1 *Note that the use of or in the representation of the Type 2 data organisation reflects the fact that the dependent variable may be represented as a $NT \times 1$ column vector indicating the index of the chosen alternative, or a $NT \times J$ matrix, with a non-zero value in the column of the chosen alternative.*

Tables 11 and 12 present two examples of data organizations using the transportation mode data set in Greene (2002). The tables give the observations for the first two individuals where the number of times periods equals 1, and there are four alternatives. Hence, since $Hinc$ denotes household income it is constant over the first and second four rows. In table 12 we illustrate the case where the dependent variable $Mode$ is defined as an index.

Table 11: Data structure Type 1: $NTJ \times K + L + 1$.

$Hinc$	$PSize$	$Ttme$	$Invc$	$Invt$	GC	$Mode$
35	1	69	59	100	70	0
35	1	34	31	372	71	0
35	1	35	25	417	70	0
35	1	0	10	180	30	1
30	2	64	58	68	68	0
30	2	44	31	354	84	0
30	2	53	25	399	85	0
30	2	0	11	255	50	1

[31] DCM will check if the indexing starts at 0 or 1.

Table 12: Data structure Type 2: $NT \times K + JL + 1$.

$Hinc$	$PSize$	$Ttme$	$Ttme$	$Ttme$	$Ttme$	...	$Mode$
35	1	69	34	35	0		4
30	2	64	44	53	0		4

DCM will determine the organization of the data from the format of the dependent variable and, conditionally on the user supplied number of time periods T, the number of individuals N and the number of alternatives.[32] DCM can handle both data structures but requires column labels that reflect the structure of the data organization. In Type 1 organization, each column label must be unique. In Type 2 organization, each column label must be identical for the J columns holding a specific characteristic or attribute (see examples in tables). Hence, since DCM needs column labels, some data *formats* are problematic if the format does not support column names.

To load the database into the object, the user calls e.g.

```
obj.Load(filename);
```

or any other `LoadXXX()`-command where `XXX` can be e.g. `Xls` or `Dht`, etc. (see Ox manual). Note that the DCM version of the `LoadXXX()` command loads the data *and* sets the database name to `filename`.[33]

Limitations in Beta version: In the current version of DCM all individuals must face the same set of alternatives.

[32] The number of time periods is set to 1 by default.

[33] The internal structure of the data is $NTJ \times K + L + 1$ for multiple index models and $NT \times JK + JL + 1$ for single index models.

B Technical appendix

B.1 Covariance matrix Specification in Multinomial Probit Models.

B.1.1 General

In Sections 3.1 and 3.4 we introduced the error covariance matrix Ξ_ε and the covariance matrix of the mean coefficients Ξ_ω. In order to impose non-negative definiteness and symmetry we parameterize the lower diagonal of the cholesky decomposition of both Ξ_ω and Ξ_ε, so that e.g. $\Xi_\omega = \mathbf{C}_\omega \mathbf{C}'_\omega$. The vectorization is performed by stacking the columns of the lower diagonal: in the case of Ξ_ω $\vec{\mathbf{C}}_\omega$ denotes a $C(C+1)/2 \times 1$ vector, where C denotes the number of mean coefficients. For $\Xi_{\varepsilon J}$ the same vectorisation results in the $J(J+1)/2 \times 1$ vector $\vec{\mathbf{C}}_\varepsilon$. Note that some elements in $\vec{\mathbf{C}}_\omega$ and $\vec{\mathbf{C}}_\varepsilon$ are fixed (not estimated).

In what follows we let Ξ_ε^* denote the unrestricted and unidentfied form of the error covariance matrix $E(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}')$; and Ξ_ε denote various just or over-identified forms.

B.1.2 The Initialisation of Ξ_ω and Ξ_ε

Initialisation of Ξ_ε: In `InitPar()` we initially set $\Xi_\varepsilon = I_J$. Depending on the assumed structure of the error covariance matrix we fix the following elements:[34]

IID: All elements in Ξ_ε are fixed. For example, in a trinomial choice model, the Ξ_ε matrix will have the following structure:

$$E(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}') = \Xi_\varepsilon = \begin{pmatrix} 1^b & . & . \\ 0^b & 1^s & . \\ 0^b & 0^* & 1^* \end{pmatrix}.$$

The superscript b indicates that the entry is fixed at this value because its the *base alternative*, s indicates that the entry is fixed because it is is the *scale alternative*, and $*$ indicates other fixed entries.

[34]This is actually done in the `SetErrDist()` procedure.

Heteroscedastic

- All elements in the rows and columns corresponding to the *reference alternative* are fixed.
  - All off-diagonal elements of the remaining $(J-1 \times J-1)$-matrix are fixed at 0.
  - The diagonal element corresponding to the *scale alternative* is fixed at 1.
  - The starting values for the remaining $J-2$ diagonal elements are set to 1.

The resulting Ξ_ε matrix is given below:

$$\Xi_\varepsilon = \begin{pmatrix} 1^b & . & . \\ 0^b & 1^s & . \\ 0^b & 0^* & 1 \end{pmatrix}.$$

Unrestricted

- All elements in the row and column corresponding to the *reference alternative* are fixed.
  - The diagonal element corresponding to the *scale alternative* is fixed at 1.
  - The starting values for the remaining $J(J-2)/2-1$ elements are set as follows: diagonal elements at 1 and off-diagonal elements at 0.

The resulting Ξ_ε matrix is given below:

$$\Xi_\varepsilon = \begin{pmatrix} 1^b & . & . \\ 0^b & 1^s & . \\ 0^b & 0 & 1 \end{pmatrix}.$$

Is this formulation of Ξ_ε identified? The most general structure of Ξ_ε in the 3 alternative case in DCM is the following:

$$E(\boldsymbol{\varepsilon}\boldsymbol{\varepsilon}') = \begin{pmatrix} 1^b & . & . \\ 0^b & 1^s & . \\ 0^b & \sigma_{23} & \sigma_{33} \end{pmatrix}. \tag{16}$$

Remark 2 *The order condition holds since we have* $J(J-1)/2 - 1 = 2$ *free covariance parameters in* Ξ_ε.

Note that if we initially take the difference of $\Xi^*_{\varepsilon J}$ with respect to alternative 2, this gives us

$$E((\varepsilon_1-\varepsilon_2)(\varepsilon_3-\varepsilon_2)') = \psi^2\Xi_{\varepsilon 3}\psi^{2'} = \Xi^2_{\varepsilon 2} = \begin{pmatrix} \sigma_{11}+\sigma_{22}-2\sigma_{12} & \sigma_{13}+\sigma_{22}-\sigma_{12}-\sigma_{23} \\ \sigma_{13}+\sigma_{22}-\sigma_{12}-\sigma_{23} & \sigma_{33}+\sigma_{22}-2\sigma_{23} \end{pmatrix}. \quad (17)$$

Imposing restrictions which are consistent with setting the reference alternative to alternative one, and the scale alternative to 2 gives us

$$\Xi^2_{\varepsilon 2} = \begin{pmatrix} 1^s + 1^b & . \\ \sigma_{32} + 1^b & \sigma_{33} + 1^b \end{pmatrix}. \quad (18)$$

By subtracting the fixed value of variance of the base alternative (1^b) from each of the entries in (18), we see that we can retrieve the entries in the original levels error covariance matrix in (16). Hence, the covariance specification is identified. The sum of the variances of the reference and scale alternative will set the overall scale of the model.

Remark 3 *Note that imposing the same normalisation (setting the reference alternative to alternative one) and scaling (setting the scale alternative to 2) will generate differenced covariance matrices* $\Xi^1_{\varepsilon 2}$ *and* $\Xi^3_{\varepsilon 2}$ *which are identical to* $\Xi^2_{\varepsilon 2}$. *This guarantees that the structure of* $\Xi^j_{\varepsilon J-1}$ *is invariant over each* j^{th} *alternative. In this regard we may also think of the identifiable set of covariance parameters as* $\varpi_{11} = 1^s + 1^b$, $\varpi_{12} = \sigma_{32} + 1^b$, *and* $\varpi_{22} = \sigma_{33} + 1^b$.

B.2 Halton Draws

Following seminal work by Bhat (1999), Bhat (2003) and McFadden and Train (2000), much of the recent emergence of the mixed logit model as a viable alternative to logit, has coincided with the use of a different approach to drawing random numbers from the unit interval. Customised simulators have been developed which in contrast to the standard type of simulator, use non-random draws from the distribution to be integrated. Pseudo Monte

Carlo methods based upon the simulation methods developed by McFadden (1989) and Pakes and Pollard (1989), are predicated on taking a set of R_i independent draws for each ith sample unit. As a result simulation errors in the criterion function are averaged out, with simulation variance decreasing at a rate of approximately $1/R$. In contrast, the approach first proposed by Bhat (1999), allocates R_i draws to each observation in a dependent sequence. Namely for observation l allocated R_l draws in the unit interval, the subsequent observation, say, m, is allocated R_m draws which fills in the gaps left by the previous observation. As a consequence, the averaging effect is more significant such that the simulation variance falls at a rate which exceeds $1/R$. There now exists a significant amount of evidence which testifies to the greater efficiency of Halton sequences, relative to the use of standard random draws (see, for example, Train (1999)). Although a number of important caveats are noted, Train also notes that in many instances the computer time required to achieve the same level of accuracy can be reduced by a factor of ten! In `DCM` the user has the option of utilising either Halton or standard random draws.