Structural choice analysis with nested logit models

Florian Heiss
University of Mannheim, MEA

Abstract. The nested logit model has become an important tool for the empirical analysis of discrete outcomes. There is some confusion about its specification of the outcome probabilities. Two major variants show up in the literature. This paper compares both and finds that one of them (called random utility maximization nested logit, RUMNL) is preferable in most situations. Since the command `nlogit` of Stata 7.0 implements the other variant (called non-normalized nested logit, NNNL), an implementation of RUMNL called `nlogitrum` is introduced. Numerous examples support and illustrate the differences between both specifications.

Keywords: st0017, nlogitdn, nlogitrum, nested logit model, discrete choice, random utility maximization model

1 Introduction

The nested logit model has become an important tool for the empirical analysis of discrete outcomes. It is attractive since it relaxes the strong assumptions of the multinomial (or conditional) logit model. At the same time, it is computationally straightforward and fast compared to the multinomial probit, mixed logit, or other even more flexible models due to the existence of a closed-form expression for the likelihood function.

There is some confusion about the specification of the outcome probabilities in nested logit models. Two substantially different formulas and many minor variations of them are presented and used in the empirical literature and in textbooks. Many researchers are neither aware of this issue nor of which version is actually implemented by the software they use. This obscures the interpretation of their results. This problem has been previously discussed by Hensher and Greene (2002), Hunt (2000), Koppelman and Wen (1998), and Louviere et al. (2000, section 6.5). This paper provides a comparison of both approaches in line with this literature. It argues and shows in numerous examples that one of these specifications is preferable in most situations. The `nlogit` command of Stata 7.0 does not implement this specification. Therefore, the `nlogitrum` command is presented, which does.

The remainder of this paper is organized as follows: Section 2 introduces basic concepts of discrete choice and random utility maximization (RUM) models and discusses the conditional logit model as the most straightforward example. Section 3 presents one version of the nested logit model, the so-called RUMNL model. It can directly be derived from a RUM model. Section 4 introduces the other variant, which is implemented as `nlogit` in Stata 7.0. It is shown that this model is more difficult to interpret and might imply counterintuitive and undesired restrictions. This is often overlooked by applied

researchers. Section 5 compares both models in special cases of nesting structures. The Stata implementation of the preferred RUMNL model is introduced in Section 6, and Section 7 concludes.

2   Fundamental concepts

Discrete choice models are used to make statistical inferences in the case of discrete dependent variables. This paper deals with a special class of discrete choice models, for which there are more than two possible outcomes that cannot be sensibly ordered. A classical example is the travel-mode choice. This paper uses a well-known dataset on this topic to provide empirical examples. Among others, Greene (2000, example 19.18), Hunt (2000), and Louviere et al. (2000, section 6.4) present nested logit estimates based on these data. The data contain 210 non-business travelers between Sydney, Canberra, and Melbourne. They had four travel-mode alternatives: car, train, bus, and plane.

Section 2.1 presents the concept of random utility maximization (RUM) models. Different types of variables can enter RUM models of discrete choice. Since this will be important for the following discussion, Section 2.2 characterizes these variable types and the specification of their coefficients. Section 2.3 presents the RUM interpretation of the well-known conditional logit model and first estimates.

2.1   Random utility maximization models

Econometricians often interpret discrete choice models in terms of underlying structural models of behavior, called random utility maximization (RUM) models. They assign a utility level U_{ij} to each alternative $j = 1, \ldots, J$ for each decision maker $i = 1, \ldots, I$. The decision makers are assumed to choose the alternative from which they derive the highest utility.

The utilities are determined by a large number of characteristics of the decision maker and the alternatives. The researchers have information on some of those determinants, but not on all. This is reflected by splitting the utilities into a deterministic part V_{ij} and a stochastic part ϵ_{ij}:

$$U_{ij} = V_{ij} + \epsilon_{ij} \tag{1}$$

The probability P_{ij} that individual i chooses some alternative j is equal to the probability of U_{ij} being the largest of all $U_{i1}, \ldots U_{iJ}$. With $y_i \in \{1 \ldots J\}$ denoting the alternative that decision maker i chooses, this probability is

$$
\begin{aligned}
P_{ij} = \Pr(y_i = j) &= \Pr(U_{ij} > U_{ik} \quad \forall\, k = 1, \ldots, J : k \neq j) \tag{2} \\
&= \Pr(\epsilon_{ik} - \epsilon_{ij} \leq V_{ij} - V_{ik} \quad \forall\, k = 1, \ldots, J : k \neq j) \tag{3}
\end{aligned}
$$

Given the deterministic parts of the utility functions $V_{i1}, \ldots, V_{iJ}$, this probability will depend on the assumptions on the distribution of the stochastic error terms $\epsilon_{i1}, \ldots, \epsilon_{iJ}$. For some distributions, there exists a closed-form solution for this expression. The most

prominent examples are the conditional logit model discussed in Section 2.3 and the random utility version of the nested logit model discussed in Section 3.2.

A look at equation (2) reveals two interesting properties of the RUM outcome probabilities: They are based on utility *differences* only. The addition of a constant to all utilities does not change the outcome probabilities. In addition to that, the scale of utility is not identified: Multiplying *each* of the utilities $U_{i1}, \ldots, U_{iJ}$ by a constant factor does not change the probabilities. So, RUM models have to normalize the utilities.

2.2 Types of variables and coefficients

The deterministic utility components V_{ij} may consist of different types of determinants. Alternative-specific constants α_j for all but one (the reference) alternative should enter the model. They capture choice probabilities *relative to the reference alternative* that cannot be attributed to the other explanatory variables. In addition, individual-specific and/or alternative-specific variables may enter the utilities.

Individual-specific variables describe characteristics of the decision maker. These variables may influence the relative attractiveness of the alternatives. Prominent examples are socio-economic variables like `income` or `age`. They are collected in a vector $\mathbf{z}_i$ for each decision maker $i = 1, \ldots, I$. A parameter vector $\boldsymbol{\gamma}_j$ for each alternative j is associated with the individual-specific variables. Since only utility *differences* are relevant for the choice, the parameters for one (the reference) alternative have to be normalized to zero for purposes of identification.[1] The other parameters can be estimated freely. They represent the effect of the individual-specific variables on the utility of the respective alternatives *relative* to the reference alternative. In the travel-mode-choice example, the respondents were asked about their household income. The individual-specific variable $\texttt{inc}_i$ represents the income of individual i in tens of thousands of dollars.

Alternative-specific variables vary both over individuals and alternatives. A prominent example is the price in models of brand choice. In the travel-mode-choice data, there is a variable $\texttt{time}_{ij}$ that represents the time (in hours) that individual i would need for the trip with travel mode j. These variables will be collected in a vector $\mathbf{x}_{ij}$ for each decision maker $i = 1, \ldots, I$ and for each alternative $j = 1, \ldots, J$. They may enter the utilities in two different ways. Since the variation over alternatives provides additional ground for identification, a separate parameter for *each* alternative is statistically identified. In the travel-mode-choice example, spending one hour in their own car might be associated with a lower disutility than spending one hour in the bus. This would be reflected in a larger $\beta_{\texttt{bus}}$ than $\beta_{\texttt{car}}$ in absolute value.

Including all these variables, the deterministic part of the utility V_{ij} can, in general, be written as

$$V_{ij} = \alpha_j + \mathbf{x}'_{ij}\boldsymbol{\beta}_j + \mathbf{z}'_i\boldsymbol{\gamma}_j \tag{4}$$

On the other hand, researchers often want to estimate a joint coefficient $\boldsymbol{\beta}$ for all

---

[1] Of course any other value can be chosen for normalization. The normalization to zero simplifies the interpretation of the other parameters.

alternatives. This is possible because of the variation of $\mathbf{x}_{ij}$ over the alternatives. In this case, we will call these variables generic variables and add the restriction

$$\boldsymbol{\beta}_j = \boldsymbol{\beta} \quad \forall j = 1, \ldots, J \tag{5}$$

With this specification, the joint parameter $\boldsymbol{\beta}$ of travel time in our example may be interpreted as the value of time in terms of utility. If price is included as a generic variable, its parameter is often used to rescale the utility in dollar terms. Whether or not generic variables enter the model will affect the discussion of the nested logit model below.

2.3   Multinomial/Conditional/McFadden's logit model

The multinomial logit (MNL) and conditional logit (CL) models are probably the most widely used tools for analyzing discrete dependent variables. The terminology is not consistent in the literature, but this paper refers to the MNL model as a special case of a CL model in which all explanatory variables are individual specific. Such a model is implemented in Stata as `mlogit`; see [R] **mlogit**. The more general conditional logit model is implemented as the `clogit` command; see [R] **clogit**. The same model without the interpretation in terms of an underlying RUM model is often referred to as multinomial logistic regression. In the following, this paper will discuss the most general CL model.

Consider a RUM model as described in Section 2.1. The CL model assumes that the error terms $\epsilon_{i1}, \ldots, \epsilon_{iJ}$ are i.i.d. as Extreme Value Type I. This distribution has a variance of $\sigma^2 = \pi^2/6$, which implicitly sets the scale of the utilities. McFadden (1974) shows that under these assumptions, the resulting probability P_{ij}^{CL} that individual $i = 1, \ldots, I$ chooses some alternative $j = 1, \ldots, J$ has a straightforward, analytical solution:

$$P_{ij}^{\text{CL}} = \frac{e^{V_{ij}}}{\sum_{k=1}^{J} e^{V_{ik}}} \tag{6}$$

Table 1 shows estimation results for two CL models of the travel mode choice example. Both consider `income` and `time` as explanatory variables and define the outcome `air` as the reference outcome; i.e., α_{air} and γ_{air} are normalized to zero. The deterministic parts of the utility in equation (4) are, therefore,

$$
\begin{aligned}
V_{i,\text{air}} &= & \beta_{\text{air}} \cdot \texttt{time}_{i,\text{air}} \\[2mm]
V_{i,\text{car}} &= \alpha_{\text{car}} & + \beta_{\text{car}} \cdot \texttt{time}_{i,\text{car}} & + \gamma_{\text{car}} \cdot \texttt{inc}_i \\[2mm]
V_{i,\text{bus}} &= \alpha_{\text{bus}} & + \beta_{\text{bus}} \cdot \texttt{time}_{i,\text{bus}} & + \gamma_{\text{bus}} \cdot \texttt{inc}_i \\[2mm]
V_{i,\text{train}} &= \alpha_{\text{train}} & + \beta_{\text{train}} \cdot \texttt{time}_{i,\text{train}} & + \gamma_{\text{train}} \cdot \texttt{inc}_i
\end{aligned}
\tag{7}
$$

for both models.

Table 1: Conditional Logit estimates

Model		(A)		(B)	
		Coef.	z	Coef.	z
const×car		−4.122	−4.09	−3.886	−3.97
	bus	−2.614	−2.33	−2.678	−2.68
	train	−1.153	−1.14	−1.523	−1.60
inc×	car	−0.209	−1.66	−0.201	−1.60
	bus	−0.454	−3.00	−0.457	−3.02
	train	−0.680	−4.92	−0.678	−4.93
time				−0.600	−8.29
time×	air	−3.364	−7.92	−2.754	−7.43
	car	−0.572	−7.58		
	bus	−0.609	−6.92		
	train	−0.639	−8.02		
Log likelihood		−201.34		−202.19	

Model A allows for different time parameters $\boldsymbol{\beta}_j$ for all alternatives. The estimates of all three $\boldsymbol{\gamma}_j$ alternatives are negative. This implies that higher income decreases the probability of choosing a travel mode other than air. The relative magnitude can also be interpreted: the order of the coefficients corresponds to the order of the marginal effects of the choice probabilities. All `time` parameters are highly and significantly negative. This implies that the time spent for the trip is associated with a disutility and that the probability of choosing any travel mode decreases as the time spent traveling increases.

As the results from model A indicate, the `time` parameters for the alternatives train, bus, and car are very similar. A test of the hypothesis that they are actually equal cannot be rejected. It makes sense to impose equality; that is, to specify `time` as a generic variable. This has two advantages. It improves the efficiency of the estimates and allows an interpretation of the coefficient as the implicit value of time in terms of utility, but $\beta_{\texttt{air}}$ is significantly higher in absolute value than the other parameters. So, model B specifies `time` as a generic variable and, additionally, includes an interaction for the `air` alternative.[2] As expected, the log-likelihood value decreases relative to the unconstrained model A, but this decrease is insignificant. The marginal effects and elasticities do not change significantly either.

The CL/MNL model is widely used because of its convenient form of the choice probabilities and due to its globally concave likelihood function that makes maximum likeli-

---

[2]There may be different reasons for the unequal parameter. Either the disutility of spending time in the plane is higher than for the other travel modes, or time actually enters the utility nonlinearly (the mean travel time by air is obviously significantly lower than the time for the other alternatives), or the situations in which people choose to fly differ in that time is more crucial. We will not further explore the reason since this paper is not really about travel mode choice and since adding nonlinear terms, etc. complicates the model unnecessarily for the purpose of demonstration.

hood estimation straightforward, but it imposes strong restrictions on the distribution of the error terms. Most notably, they are assumed to be independently distributed. Note that these terms capture all unobserved determinants of the choices. If two alternatives are similar, it is plausible to assume that their errors are positively correlated. In our example, if there are unobserved individual characteristics that affect the utility of both public transportation modes bus and train similarly, the error terms of those alternatives are correlated. This is ruled out by the CL model. If the assumption of independent error terms is violated, the CL parameter estimates are biased.

3   Nested logit models I: RUMNL

The basic idea of nested multinomial logit (NMNL) models is to extend the CL model in order to allow groups of alternatives to be similar to each other in an unobserved way; that is, to have correlated error terms. The general approach of NMNL models is introduced in Section 3.1. Section 3.2 presents a NMNL model that is derived from a RUM model and is therefore called RUMNL model in this paper. Finally, Section 3.3 extends the CL example for this model. A Stata implementation of the RUMNL model is introduced later in this paper; see Section 6.

3.1   General approach

The researcher partitions the choice set into M subsets ('nests') $B_m, m = 1, \ldots, M$[3], so that each alternative belongs to exactly one nest. Denote the nest to which alternative $j = 1, \ldots, J$ belongs as $B(j)$:

$$B(j) = \{B_m : j \in B_m, \quad m = 1, \ldots M\} \tag{8}$$

For the travel mode example, one possible nesting structure is depicted in Figure 1. The number of nests is $M = 2$. The public transportation modes (train and bus) share the nest $B_{\text{public}} = \{\text{bus}, \text{train}\}$, and the other modes (air and car) share the nest $B_{\text{other}} = \{\text{car}, \text{air}\}$. In our notation, $B(\text{bus})$ is equivalent to B_{public} just as $B(\text{train})$ is. This notation will help in formulating the choice probabilities below.

<div align="center">

(*Continued on next page*)

</div>

---

[3]This can be generalized to various nesting levels in a straightforward way by grouping the alternatives within such a nest in sub-nests and so on, but we will concentrate on the simplest case of only one nesting level.
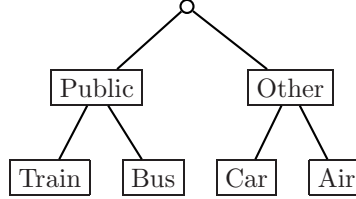
Figure 1: Nesting structure for models C through H

In order to develop an intuitive expression for the choice probabilities, it helps to decompose them into two parts. The probability of individual i choosing alternative j, $\Pr(y_i = j)$, is equal to the product of the probability to choose some alternative in nest $B(j)$, $\Pr\{y_i \in B(j)\}$, and the conditional probability to choose exactly alternative j *given* some alternative in the same nest $B(j)$ is chosen $\Pr\{y_i = j|y_i \in B(j)\}$; that is,

$$P_j = \Pr(y = j) = \Pr\{y = j|y \in B(j)\} \cdot \Pr\{y \in B(j)\} \tag{9}$$

where the individual subscript i is dropped from now on for the sake of a more concise notation. In our example, the probability of taking the bus $\Pr(y = \texttt{bus})$ is equal to the probability of choosing public transportation $\Pr\{y \in B(\texttt{bus})\}$ times the conditional probability of taking the bus *given* a public transportation mode is chosen $\Pr\{y = \texttt{bus}|y \in B(\texttt{bus})\}$. Note that this decomposition is valid in general by the rules of conditional probability, but it is especially useful for thinking about the nested logit model.

3.2   Nested logit as a RUM model

The NMNL model can be derived from a RUM model just as the CL model. Consider a RUM model as described in 2.1. The CL model assumes that the error terms $\epsilon_{i1}, \ldots, \epsilon_{iJ}$ are i.i.d. as Extreme Value Type I. Instead, the RUMNL model assumes a generalized version of this distribution. This special form of the generalized extreme value (GEV) distribution extends the Extreme Value Type I distribution by allowing the alternatives within a nest to have mutually correlated error terms.

For each nest $m = 1, \ldots, M$, the joint distribution of the error terms has an additional parameter τ_m that represents a measure of the mutual correlation of the error terms of all alternatives within this nest. Actually, this paper specifies τ_m to be equal to $\sqrt{1 - \rho_m}$, with ρ_m representing the correlation coefficient. So, it is an *inverse* measure of the correlation. Therefore, it is often called dissimilarity parameter.[4] The *marginal* distribution of each error term is again Extreme Value Type I.

The RUMNL conditional choice probability of choosing alternative j given some alternative in its nest is chosen is $\Pr\{y = j|y \in B(j)\}$, which corresponds to a simple CL model for the choice between the alternatives in nest $B(j)$. The utilities are rescaled by

---

[4]Other equivalent parameterizations are used in the literature. For example, McFadden (1981) replaces τ_m with $\sigma_m = 1 - \tau_m$, and Louviere et al. (2000) replace τ_m with $\mu_m = 1/\tau_m$.

the inverse of the dissimilarity parameter $\tau(j)$ for this nest:

$$\Pr\{y = j | y \in B(j)\} = \frac{e^{V_j/\tau(j)}}{\sum_{k \in B(j)} e^{V_k/\tau(j)}} \tag{10}$$

The most intuitive explanation is based on the consideration of the implicit scaling in the logit model. As seen in Section 2.1, the RUM choice probabilities depend on the utility *differences*. As noted above, the CL model implicitly scales all utilities such that the error terms have a variance of $\sigma^2 = \pi^2/6$. Since they are assumed to be independent in the CL model, their differences have a variance of $2\sigma^2$, but the RUMNL error terms within a nest are positively correlated. The higher the correlation between the error terms, the lower is the variance of these differences. With the relationship between the dissimilarity parameter τ_m and the coefficient of correlation ρ_m presented above, it is straightforward to show that the variance of the difference is $2\sigma^2\tau_m^2$. By normalizing the utilities by the factor $1/\tau_m$, the variance of this normalized difference becomes $2\sigma^2$. Without this normalization, the utilities in each nest would be scaled by a different factor and would therefore not be comparable across nests.

The denominator in equation (10) represents a (rescaled) measure of the attractiveness of the nest $B(j)$. The log of this expression for each nest m is called inclusive value IV_m. It corresponds to the expected value of the utility individual i obtains from the alternatives in nest m:

$$IV_m = \ln \sum_{k \in B_m} e^{V_k/\tau_m} \tag{11}$$

The probability $\Pr\{y \in B(j)\}$ of choosing some alternative from nest k is again a CL probability for the choice between the nests. The scaled back inclusive values take the role of the deterministic parts of the utilities:

$$\Pr\{y \in B(j)\} = \frac{e^{\tau(j)IV(j)}}{\sum_{m=1}^{M} e^{\tau_m IV_m}} \tag{12}$$

Because of the way the dissimilarity parameters enter this equation, they are also called IV parameters.

Nested logit models can be fit sequentially. First, fit a sub-model for each nest according to equation (10). Then, calculate the inclusive values defined in equation (11) and fit a model for the choice of a nest shown in equation (12). See, among others, Train (2002) for a discussion of this sequential estimation and the necessary decomposition of the explanatory variable into nest- and alternative-specific variables. Alternatively, all these equations can be plugged into equation (9). In this way, we obtain the marginal choice probability for alternative j as

$$P_j^{\mathrm{RNL}} = \frac{e^{V_j/\tau(j)}}{e^{IV(j)}} \times \frac{e^{\tau(j)IV(j)}}{\sum_{m=1}^{M} e^{\tau_m IV_m}} \tag{13}$$

This probability is the full information likelihood contribution.

The CL model follows as a special case when $\tau_m = 1$, $\forall m = 1, \ldots, M$. This can be easily checked: the nests merely partition the choice set, so $\sum_{m=1}^{M} e^{IV_m} = \sum_{k=1}^{J} e^{V_k}$ must hold in this case. The RUMNL model is consistent with RUM if all τ_m lie in the unit interval.[5] For an introduction to this model, also see Train (2002) and Maddala (1983).

3.3 Examples

Table 2 shows estimation results for two RUMNL models of the travel-mode-choice example with the nesting structure depicted in Figure 1. Model C corresponds to a RUMNL version of the CL model A. The log-likelihood value increases considerably by allowing the IV parameters to diverge from unity. A likelihood-ratio test clearly rejects the CL model that implicitly restricts the IV parameters to unity. The IV parameter $\tau_{\texttt{public}}$ is within the unit interval and corresponds to a correlation of the two error terms of about .71. The IV parameter $\tau_{\texttt{other}}$ is clearly above 1. This implies that this model is inconsistent with RUM. We will ignore this for now and come back to this issue in Section 5.2.

Table 2: RUMNL estimates

Model		(C)		(D)	
		Coef.	z	Coef.	z
const×car		−5.751	−1.60	−6.383	−2.24
	bus	−2.499	−0.76	−2.782	−1.03
	train	−1.253	−0.39	−1.786	−0.66
inc×	car	−0.354	−0.90	−0.362	−0.93
	bus	−0.556	−1.94	−0.554	−1.93
	train	−0.827	−2.90	−0.831	−2.91
time				−1.301	−5.6
time×	air	−7.027	−5.49	−5.878	−5.54
	car	−1.325	−5.12		
	bus	−1.281	−5.37		
	train	−1.305	−5.54		
τ public		0.539	3.69	0.545	3.79
τ other		4.879	3.58	4.801	3.84
Log likelihood		−165.12		−165.26	

The other parameters tend to be larger in the RUMNL model than in the CL model. They cannot be compared, however, since the scaling differs across the models. One can either compare ratios of coefficients or calculate statistics such as the estimated marginal effects or the elasticities of the choice probabilities with respect to the explanatory

[5]This condition can be relaxed for *local* consistency with RUM, see Börsch-Supan (1990).

variables. The interpretation within the RUMNL model is equivalent to the interpretation in the CL model A. Model D in Table 2 shows a RUMNL model with time entering as a generic variable analogous to the CL model B. Again, the interpretation remains the same. The generic restrictions of model D cannot be rejected by a likelihood-ratio test.

4   Nested logit models II: NNNL

This section discusses a variant of the nested logit model. It will be called non-normalized nested logit (NNNL) model for reasons that are explained below. This is the model that is presented as the nested logit model, for example, presented by Greene (2000, section 19.7.4). It is also the model implemented in Stata 7.0 by the command nlogit; see [R] **nlogit**.

4.1   Structure of the model

A latent variable $\widetilde{V}_j$ similar to the deterministic part of the utility in a RUM model is defined as a linear combination of the explanatory variables:

$$\widetilde{V}_j = \widetilde{\alpha}_j + \mathbf{x}'_j \widetilde{\boldsymbol{\beta}}_j + \mathbf{z}' \widetilde{\boldsymbol{\gamma}}_j \tag{14}$$

If alternative-specific variables enter the model as generic variables, that is, with a common coefficient $\widetilde{\boldsymbol{\beta}}_j$ for all alternatives, analogous restrictions to equation (5) are imposed:

$$\widetilde{\boldsymbol{\beta}}_j = \widetilde{\boldsymbol{\beta}} \quad \forall j = 1, \dots, J \tag{15}$$

The reason for adding the tilde to the V and the parameters is that the variable V_j is reserved to represent deterministic utility parts in this paper, and as will be explained below, this linear combination $\widetilde{V}_j$ may not be interpreted in this way.

With the inclusive value for any nest m defined as

$$\widetilde{IV}_m = \ln \sum_{k \in B_m} e^{\widetilde{V}_k} \tag{16}$$

the choice probabilities of the NNNL model are

$$P_j^{\mathrm{NNL}} = \frac{e^{\widetilde{V}_j}}{e^{\widetilde{IV}(j)}} \times \frac{e^{\tau(j)\widetilde{IV}(j)}}{\sum_{m=1}^{M} e^{\tau_m \widetilde{IV}_m}} \tag{17}$$

Comparing these equations to equations (11) and (13), the relevant difference is that the deterministic utilities are not scaled by the inverse of the IV parameter in the conditional probability within the nest, $e^{\widetilde{V}_j}/e^{\widetilde{IV}(j)}$. This is the reason for calling this model a non-normalized nested logit (NNNL) model. As argued in Section 3.2, this implies different scaling of the utilities across nests. In consequence, the interpretation of this model as a RUM model with the deterministic utility defined as $\widetilde{V}_j$ is challenged. This can be confirmed formally by considering what happens in a RUM model when the utility

of each alternative is increased by some value a. According to Section 2.1, the RUM choice probabilities do not change. Now have a closer look at equation (16). Adding the constant a to every $\widetilde{V}_j$ *does* alter the NNNL choice probabilities.

As a result, this model is not based on a RUM model with the deterministic parts of the utilities defined as $\widetilde{V}_j$ as was noted by Hensher and Greene (2002), Hunt (2000), Koppelman and Wen (1998), and Louviere et al. (2000, section 6.5). The next section, however, argues that it *can* be interpreted in RUM terms with other deterministic utilities.

4.2 Interpretation of the NNNL as a RUM model

As a result of the discussion above, the parameters $\widetilde{\alpha}_j, \widetilde{\boldsymbol{\beta}}_j$, and $\widetilde{\boldsymbol{\gamma}}_j$ of a NNNL model may not be interpreted as the structural parameters of an underlying RUM model as many researchers tend to do. But how can the parameters be interpreted? A reformulation of the NNNL model that is motivated from the insights of Section 4.1 helps to answer this question. Suppose the deterministic part of the utility is not defined as $\widetilde{V}_j$ but as a scaled version V_j^{NNL} of it,

$$V_j^{\text{NNL}} = \tau(j)\widetilde{V}_j = \tau(j)\left(\widetilde{\alpha}_j + \mathbf{x}_j'\widetilde{\boldsymbol{\beta}}_j + \mathbf{z}'\widetilde{\boldsymbol{\gamma}}_j\right) \tag{18}$$

where $\tau(j)$ is the IV parameter of the nest to which alternative j belongs. Adding the constant a to every V_j^{NNL} means adding $a/\tau(j)$ to $\widetilde{V}_j$ and the inclusive value $\widetilde{IV}(j)$. As can be easily seen from equation (17), this leaves the choice probabilities unchanged.

If $\widetilde{V}_j$ in equations (16) and (17) are replaced with the equivalent term $a/\tau(j)V_j^{\text{NNL}}$, the equations become equivalent to the RUMNL equations (11) and (13). So, the difference between the NNNL and the RUMNL model boils down to the specification of the utilities. While the RUMNL model directly considers the deterministic utilities and their parameters α, $\boldsymbol{\beta}$, and $\boldsymbol{\gamma}$, the NNNL model specifies utility according to equation (18).

A researcher with access to NNNL software but not to RUMNL software can apply a NNNL model and deduce the implicit RUM assumptions and parameters according to equation (18). Depending on the nesting structure and the presence of generic variables, this can be more or less straightforward and more or less sensible. In some cases, the NNNL parameters "only" have to be rescaled to recover the RUM parameters. In other cases, the NNNL model implicitly imposes restrictions that are usually undesired and unnoticed by researchers and readers of their work. The next sections identify these cases in order to illustrate the theoretical arguments and to provide a guideline of how to interpret NNNL results.

4.3 Example 1: Alternative-specific coefficients only

In many applications, no generic variables enter the model. This case will turn out to be the least problematic for NNNL estimation in the sense that no implicit restrictions are imposed and the utility parameters can be recovered easily from the estimates.

The NNNL utility from equation (18) can be rewritten as

$$V_j^{\text{NNL}} \;=\; \tau(j)\left(\widetilde{\alpha}_j + \mathbf{x}_j'\widetilde{\boldsymbol{\beta}}_j + \mathbf{z}'\widetilde{\boldsymbol{\gamma}}_j\right) \tag{19}$$

$$\;=\; \tau(j)\widetilde{\alpha}_j + \mathbf{x}_j'\{\tau(j)\widetilde{\boldsymbol{\beta}}_j\} + \mathbf{z}'\{\tau(j)\widetilde{\boldsymbol{\gamma}}_j\} \tag{20}$$

So, with

$$\begin{aligned}
\alpha_j &:= \tau(j)\widetilde{\alpha}_j \\
\boldsymbol{\beta}_j &:= \tau(j)\widetilde{\boldsymbol{\beta}}_j \\
\boldsymbol{\gamma}_j &:= \tau(j)\widetilde{\boldsymbol{\gamma}}_j
\end{aligned} \tag{21}$$

the utility simplifies to the equivalent of the RUMNL specification (4):

$$V_j^{\text{NNL}} = V_j = \alpha_j + \mathbf{x}_j'\boldsymbol{\beta}_j + \mathbf{z}'\boldsymbol{\gamma}_j \tag{22}$$

So, assume that you have estimated a nested logit model using NNNL software such as the `nlogit` command of Stata. The estimates of $\widetilde{\alpha}_j$, $\widetilde{\boldsymbol{\beta}}_j$, and $\widetilde{\boldsymbol{\gamma}}_j$ do not directly have a structural interpretation in terms of a RUM model, but the underlying parameters α_j, $\boldsymbol{\beta}_j$, and $\boldsymbol{\gamma}_j$ can be recovered according to equation (21).

As an illustration, Table 3 shows the results for a NNNL (model E) that corresponds to the RUMNL model C. Both models are equivalent in terms of the log likelihood and the implied marginal effects and elasticities. The estimated IV parameters are also identical, but the other parameter estimates differ. For example, the RUMNL estimate for the structural parameter of 'inc×train' is $\widehat{\gamma}_{\text{inc}\times\text{train}} = -0.474$. It can be recovered from the NNNL estimates by multiplying the estimated coefficient $\widehat{\widetilde{\gamma}}_{\text{inc}\times\text{train}} = -0.879$ with the estimated IV parameter of the respective nest $\widehat{\tau}_{\text{public}} = 0.539$, as can be easily verified: $-0.474 = -0.879 \times 0.539$. Table 3 does these calculations for each of the coefficients. The third column shows the scaling factors, which correspond to the respective estimated IV parameter. The products of these factors and the estimated NNNL coefficient can be found in the fourth column of Table 3. Their equality to the RUMNL parameters can be easily verified by a comparison with the RUMNL results in the first column of Table 2.

Table 3: NNNL estimates without generic variables

		Model (E)		Recovering RUM params		
		Coef.	z	Factor	Par.	z^*
const×car		-1.179	-1.29	4.879	-5.751	1.60
	bus	-4.635	-0.73	0.539	-2.499	0.77
	train	-2.323	-0.38	0.539	-1.253	0.40
inc×	car	-0.072	-0.90	4.879	-0.354	0.90
	bus	-1.031	-1.82	0.539	-0.556	1.94
	train	-1.534	-2.48	0.539	-0.827	2.90
time×	air	-1.440	-3.63	4.879	-7.027	5.49
	car	-0.272	-5.03	4.879	-1.325	5.12
	bus	-2.376	-4.92	0.539	-1.281	5.37
	train	-2.420	-4.87	0.539	-1.305	5.54
τ public		0.539	3.69			
τ other		4.879	3.58			
Log likelihood		-165.12				

*: Wald test of H_0: Rescaled parameter $= 0$. Shown is the
square root of the test statistic asymptotically $N(0,1)$

Note that the NNNL parameters cannot be interpreted in terms of RUM directly. The relative size of the coefficients does not have any meaning before they are rescaled. The scaling also has to be taken into account when testing hypotheses based on the parameters. For example, the presented asymptotic t statistic for $\widetilde{\gamma}_{\text{inc×train}}$ for the NNNL model does not correspond to the respective test for the RUMNL parameter $\gamma_{\text{inc×train}}$. The tests of the RUMNL parameters can be reproduced from the NNNL estimates. The appropriate null hypothesis $H_0 : \widetilde{\gamma}_{\text{inc×train}} \times \tau_{\text{public}} = 0$ can, for example, be tested using a Wald test. The respective test statistics for all parameters are shown in the fifth column of Table 3. They are equivalent to the asymptotic t statistics of the RUMNL model (C).[6]

So, in the case without generic variables, the NNNL and RUMNL models are equivalent. But while the RUMNL model directly estimates the parameters of interest, the estimated coefficients from NNNL have to be rescaled before they can be interpreted. This rescaling also has to be taken into account when testing hypotheses. For example, the asymptotic t statistics from the output of `nlogit` do not correspond to tests of intrinsically interesting hypotheses.

---

[6] The test statistic for the Wald test is asymptotically χ_1^2. The displayed value is the square root of this statistic, which is asymptotically $N(0,1)$ by the properties of the χ^2 distribution with one d.f.

4.4   Example 2: Inclusion of generic variables

As discussed above, the researcher may often want to constrain the coefficients $\boldsymbol{\beta}_j$ of alternative-specific variables to be equal for each alternative. This constraint, (5): $\boldsymbol{\beta}_j = \boldsymbol{\beta} \quad \forall j = 1, \dots, J$, could easily be imposed for the CL model B and for the RUMNL model D. However, the corresponding constraints on the NNNL parameters according to equation (15) are *not* equivalent. Instead of equal RUM parameters, they impose equal *scaled* RUM parameters:

$$\widetilde{\boldsymbol{\beta}}_j = \widetilde{\boldsymbol{\beta}} \quad \forall j = 1, \dots, J \tag{23}$$

$$\Leftrightarrow \quad \frac{1}{\tau(j)} \boldsymbol{\beta}_j = \widetilde{\boldsymbol{\beta}} \quad \forall j = 1, \dots, J \tag{24}$$

$$\Leftrightarrow \quad \boldsymbol{\beta}_j = \tau(j) \widetilde{\boldsymbol{\beta}} \quad \forall j = 1, \dots, J \tag{25}$$

The structural parameters $\boldsymbol{\beta}_j$ are not restricted to be equal across alternatives. Instead, they are constrained to be proportional to the IV parameters of their nest. The author of this paper cannot think of a RUM model for which these constraints could make any sense. Why should the travel time in our example be associated with more disutility for travel modes that happen to share a nest with relatively dissimilar alternatives?

Table 4 shows NNNL estimates with `time` specified as a "generic" variable in the sense of equation (23). A comparison of models F and D illustrates that the NNNL model does not give the same estimates as the RUMNL model in this case. In particular, the log-likelihood values differ. While the corresponding RUMNL model D shows very different IV parameters for both nests (0.55 vs. 4.80), the estimates of the NNNL IV parameters from model F are relatively similar for both nests (2.54 vs. 2.64). With the intuition developed so far, this can be readily interpreted. In the RUMNL model, the IV parameters solely capture the (dis)similarity of the alternatives within the corresponding nest. While the public transportation modes appear to be quite similar, the other modes are not. This is reflected in the RUMNL estimates. The IV parameters in the NNNL model capture another effect: the relative importance of travel time for the alternatives within the nest. The diverging IV parameters that are in accordance with the dissimilarity would imply that travel time is much more important for the car alternative than for the public transportation modes. This is not the case, as is obvious from the previous results. So, both effects that are captured by the same NNNL IV parameters are not in line with each other.

(Continued on next page)

Table 4: NNNL estimates with generic variables

Model		(F) NNNL		(G) NNNL		(H) RUMNL	
		Coef.	z	Coef.	z	Coef.	z
const×car		−2.325	−2.56	−2.556	−3.01	−6.645	−3.26
	bus	−2.364	−2.87	−2.398	−3.03	−6.235	−2.88
	train	−1.319	−1.73	−1.358	−1.86	−3.531	−1.89
inc×	car	−0.138	−1.34	−0.150	−1.47	−0.390	−1.47
	bus	−0.196	−1.56	−0.191	−1.54	−0.497	−1.64
	train	−0.352	−3.18	−0.349	−3.24	−0.907	−3.68
time		−0.460	−6.75	−0.456	−6.73	−1.185	−5.64
time× air		−1.988	−5.39	−2.079	−6.04	−5.405	−5.46
τ public		2.535	4.29	2.600	4.41	2.600	4.41
τ other		2.638	4.36	2.600	4.41	2.600	4.41
Log likelihood		−194.01		−194.29		−194.29	

The "generic" specification for the NNNL model implies a counterintuitive restriction that can hardly be motivated from a RUM model. As a result, specifications like model F should be avoided. RUM models like model D can, in general, not be estimated with NNNL software like Stata's `nlogit` command if generic variables are present. There are exceptions, some of which are discussed in the next section.

5 Special nesting structures

Section 4.4 argued, that the specification of NNNL models with generic variables can, in general, imply implausible binding constraints. This section discusses special cases for which this is not true.

5.1 Equal IV parameters across all nests

If one is willing to assume a priori that the dissimilarity parameters of all nests in a nesting level have the same value, the scaling problem of the NNNL model disappears. The restrictions (23) imply essentially the same as the generic restrictions in a RUMNL model according to equation 5. The presence of the generic variable does not distort the estimates of the NNNL model, since its parameter is forced to be scaled equally in each nest.

Table 4 shows results for a NNNL and a RUMNL model that differ from the previous ones in that their IV parameters are constrained to be equal. The RUMNL parameters (model H) can be deduced from the NNNL estimates by multiplying them with the

joint IV parameter. For example, the estimated RUMNL income coefficient for the train alternative is $\widehat{\gamma}_{\text{inc}\times\text{train}} = -0.907$. It can be recovered from the NNNL estimates as $\widehat{\widehat{\gamma}}_{\text{inc}\times\text{train}} \times \widehat{\tau}_{\text{public}} = -0.349 \times 2.600$.

The problem with this constraint is that it cannot be tested with NNNL estimates, because the unconstrained model F is misspecified. In contrast, both RUMNL specifications are valid, and a comparison of the log-likelihood values of models D and H clearly shows that this constraint is rejected by the data.

5.2   Degenerate nests

If a nest contains only one alternative, it is called a degenerate nest. The dissimilarity parameter of degenerate nests is not defined in the RUMNL model. This can be easily seen from equations (11) and (13). Since the degenerate nest $B(j)$ only contains alternative j, its inclusive value (11) simplifies to $IV(j) = V_j/\tau(j)$. The dissimilarity parameter $\tau(j)$ cancels out of the choice probability (13). This is intuitive since the concept of (dis)similarity does not make sense with only one alternative.

In the NNNL model, however, the dissimilarity parameter of degenerate nests does not vanish from the choice probability and may be statistically identified. As discussed above, the identification in general comes from two sources: the dissimilarity and the relative importance of the "generic" variables in the respective nest. Like in the RUMNL model, the former source disappears in degenerate nests, but the latter source may be present if generic variables enter the model. Without "generic" variables, the dissimilarity parameters are not jointly identified with the other parameters, so they can be constrained to any nonzero value. The only effect of choosing this value is that the respective parameters are scaled accordingly as discussed in Section 4.3.

If at least one "generic" variable is included in the NNNL model, the IV parameter of degenerate nests may be identified along with the other model parameters. This identification comes from the restriction of equally *scaled* parameters $\boldsymbol{\beta}_j/\tau(j)$ across alternatives and nests, and the parameters only constitute this scaling. A conventional approach to restrict the IV parameter to be equal to unity does not result in a model that is consistent with the underlying RUM model.

This is demonstrated with the estimates shown in Table 5. The fact that the estimated dissimilarity parameter of the nest `other` in Table 2 is substantially larger than 1 indicates that the alternatives `air` and `car` should not share a nest. Therefore, the nesting structure is modified by splitting this nest into two degenerate nests. The resulting nesting structure is depicted in Figure 2. In models I and J shown in Table 5, the variable `time` purely enters as a generic variable. The dissimilarity parameters of the degenerate nests `air` and `car` are not identified from the RUMNL model I. As argued above, they cancel out in the likelihood function. In contrast, all IV parameters are identified in the NNNL model J. It has two more free parameters than the RUMNL model and a substantially higher likelihood value.
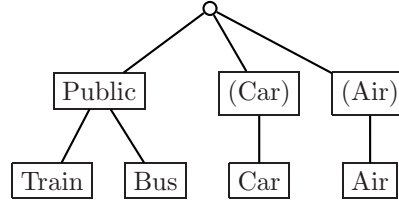
Figure 2: Nesting structure for models I through L

Table 5: Degenerate Nests

Model	(I) RUMNL		(J) NNNL		(K) RUMNL		(L) NNNL	
	Coef.	z	Coef.	z	Coef.	z	Coef.	z
const×car	1.140	1.97	−19.400	−2.74	−3.613	−3.83	−3.613	−3.83
bus	3.206	6.17	−7.283	−1.48	−1.433	−1.56	−7.283	−1.48
train	3.371	6.19	−5.130	−1.07	−1.010	−1.11	−5.130	−1.07
inc× car	−0.011	−0.10	−0.695	−1.09	−0.130	−1.09	−0.130	−1.09
bus	−0.451	−4.31	−2.328	−2.74	−0.458	−3.81	−2.328	−2.74
train	−0.505	−4.83	−3.013	−3.13	−0.593	−4.86	−3.013	−3.13
time	−0.165	−3.79	−2.319	−4.66				
time× public					−0.456	−6.17	−2.319	−4.66
air					−2.654	−6.73	−2.654	−6.73
car					−0.432	−6.11	−0.432	−6.11
τ public	0.073	2.96	0.197	3.78	0.197	3.78	0.197	3.78
τ air	—	—*	1.144	3.86	—	—*	1	—**
τ car	—	—*	0.186	3.74	—	—*	1	—**
Log likelihood	−212.45		−182.57		−182.57		−182.57	

*: Parameter not defined.

**: Parameter normalized to 1.

However, these IV parameters do not have anything to do with (dis)similarity. They simply relax the constraint of equal scaling of the generic variable coefficient across nests. To demonstrate this, models K and L shown in Table 5 do the same explicitly by estimating a separate `time` coefficient for each nest. As a result, the IV parameters of the degenerate nests are not jointly identified with the other parameters of the corresponding nests in the NNNL model and have to be constrained to any nonzero number. Both models result in the same log-likelihood value and the parameters are equivalent if the NNNL parameters are rescaled with the value of the corresponding IV parameter. The results are also equivalent to model J. This supports the assertion that the IV parameters in model J do nothing more than relax the constraint of equal scaling.

So, if there is only one generic variable present in the model, the NNNL estimate of the IV parameter can be interpreted in a straightforward way, although this is probably not the way the researcher intends to interpret IV parameters. It is much more direct to explicitly relax the specification of generic variables. If there is more than one generic variable, the interpretation becomes more obscure. Then, the NNNL specification imposes the restriction that the coefficients of all generic variables differ proportionally across nests. Greene (2000, example 19.18) presents a model in which this problem appears. It is a NNNL model based on the data used in this paper. In addition to `time`, the generic variable `cost` is included. As a result, the estimates have no clear interpretation. The RUMNL avoids the danger of misspecification and misinterpretation.

5.3   Dummy nests

There is a way to trick NNNL software into estimating a RUM consistent nested logit model with generic variables and without imposing equality of dissimilarity parameters. Koppelman and Wen (1998) propose to add degenerate dummy nests and constrain their IV parameters appropriately. This can most easily be explained by an example.

Figure 3 shows the nesting structure for the travel-mode-choice example according to Figure 1 with appropriate dummy nests added. For each alternative, such a degenerate nest is specified. The corresponding IV parameters θ_1 through θ_4 are shown next to each nest along with the respective constraint. The two `public` alternatives each have a degenerate dummy nest whose IV parameters are constrained to be equal to the IV parameter of the `other` nest. Intuitively, their parameters are first scaled by $1/\tau_{\texttt{public}}$. Then the additional dummy nest scales them by $1/\tau_{\texttt{other}}$. For the two 'other' alternatives, this works accordingly. As a result, the parameters of *all* alternatives are scaled by $1/(\tau_{\texttt{public}}\tau_{\texttt{other}})$. While τ_1 and τ_2 can be allowed to differ, this does not translate into different scaling across nests.
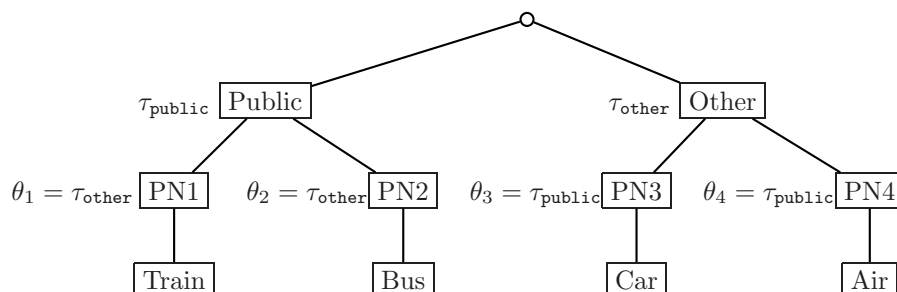
(*Continued on next page*)

Figure 3: Nesting structure with dummy nests

Table 6 shows the results from a specification according to this strategy. Model M is identical to model D. It could not be reproduced by NNNL since it contains generic variables and the IV parameters are allowed to differ between nests. Model N is a NNNL model with the dummy nests added as described above (NNNL-DN). As can be seen, this specification mimics the RUMNL model except for the scaling of the parameters for the explanatory variables. The structural coefficients can be recovered from these estimates by multiplying the estimated coefficients by both estimated IV parameters. For example, the coefficient for inc×train is -0.831. It can be calculated from the NNNL-DN estimates as $-0.831 = -0.317 \times 4.801 \times 0.545$.

Table 6: Dummy nests

Model		(M)=(D) RUMNL		(N) NNNL-DN	
		Coef.	z	Coef.	z
const×car		-6.383	-2.24	-2.438	-1.51
	bus	-2.782	-1.03	-1.063	-0.86
	train	-1.786	-0.66	-0.682	-0.59
inc×	car	-0.362	-0.93	-0.138	-0.92
	bus	-0.554	-1.93	-0.212	-1.58
	train	-0.831	-2.91	-0.317	-1.98
time		-1.301	-5.60	-0.497	-3.08
time× air		-5.878	-5.54	-2.245	-2.69
τ public		0.545	3.79	0.545	3.79
τ other		4.801	3.84	4.801	3.84
Log likelihood		-165.257		-165.26	

Depending on the original nesting structure, a large number of dummy nests may be needed for this strategy. This complicates both the specification and the estimation.[7]

---

[7]The command `nlogitdn`, introduced in Section 6.7, automates the generation of dummy nests and appropriate constraints.

Therefore, this strategy seems to be a real alternative to RUMNL only for researchers who just have access to a NNNL implementation.

6   Stata implementation of RUMNL

The NNNL model is available for Stata 7.0 users as the `nlogit` command. As argued in this paper, the RUMNL model is preferable in most situations. This section introduces the command `nlogitrum.ado` that implements the RUMNL model. It was used to produce all RUMNL estimates in this paper. Furthermore, the command `nlogitdn.ado` is described. It adds dummy nests to any specified nesting structure, as discussed in Section 5.3.

6.1   Syntax

`nlogitrum` *depvar indepvars* [*weight*] [`if` *exp*] [`in` *range*] `,` `group`(*varname*)
    `nests`(*altsetvarB* [... *altsetvar2 altsetvar1*]) [<u>notr</u>ee <u>nola</u>bel <u>clo</u>git
    <u>level</u>(*#*) <u>nolog</u> <u>robust</u> <u>ivc</u>onstraints(*string*) <u>constr</u>aints(*numlist*)
    maximize_options]

`by` `...:` may be used with `nlogitrum`; see [R] **by**.

`fweight`s and `iweight`s are allowed; see [U] **14.1.6 weight**, but they are interpreted to apply to groups as a whole and not to individual observations.

`nlogitrum` shares the features of all estimation commands; see [U] **23 Estimation and post-estimation commands**.

where

depvar	is a dichotomous variable coded as 0 for not selected alternatives and 1 for the selected alternative.
indepvars	are the attributes of the bottom-level alternatives (absolute or perceived) and possibly interactions of individual attributes with the bottom-level alternatives.
altsetvarB	is a categorical variable that identifies the bottom, or final, set of all alternatives.
altsetvar2	is a categorical variable that identifies the second-level set of alternatives—these must be mutually exclusive groups of the third-level alternatives.
altsetvar1	is a categorical variable that identifies the top- or first-level set of alternatives—these alternatives must be mutually exclusive groups of the second-level alternatives.

6.2 Syntax for predict

`predict` $\begin{bmatrix} type \end{bmatrix}$ *newvarname* $\begin{bmatrix} \texttt{if} \ exp \end{bmatrix}$ $\begin{bmatrix} \texttt{in} \ range \end{bmatrix}$ $\begin{bmatrix} , \ statistic \end{bmatrix}$

where *statistic* is

pb   predicted probability of choosing bottom-level, or choice-set,
        alternatives—each alternative identified by *altsetvarB*; the default.
p1   predicted probability of choosing first-level alternatives—each
        alternative identified by *altsetvar1*.
p2   predicted probability of choosing second-level alternatives—each choice
        identified by *altsetvar2*.
...
p#   predicted probability of choosing #-level
        alternatives—each alternative identified by *altsetvar#*.
xb   linear prediction for the bottom-level alternatives.
condpb Pr(each bottom alternative | alternative is available after all earlier choices).
condp1 Pr(each level 1 alternative) = p1.
condp2 Pr(each level 2 alternative | alternative is available after level 1 decision).
condp3 Pr(each level 3 alternative | alternative is available after stage 1
        and stage 2 decisions).
...
condp# Pr(each level # alternative | alternative is
        available after all previous stage decisions).
ivb   inclusive value for the bottom-level alternatives.
iv1   inclusive value for the first-level alternatives.
iv2   inclusive value for the second-level alternatives.
...
iv#   inclusive value for the #-level alternatives.

The inclusive value for the first-level alternatives is not used in the estimation of the model,
    therefore, it is not calculated.

These statistics are available both in and out of sample; type `predict ... if e(sample) ...`
    if wanted only for the estimation sample.

6.3 Data setup

The data setup for `nlogitrum` is equivalent to `nlogit`. That is, a set of categorical variables *altsetvarB* [... *altsetvar2 altsetvar1*] is generated using `nlogitgen`. The tree structure can be visualized using `nlogittree`. For a thorough description, see [R] **nlogit**.

The syntax is similar to that of `nlogit`, with one major difference. `nlogit` insists on explanatory variables for each nesting level, and `nlogitrum` only allows explanatory variables to directly enter the conditional probabilities of the alternatives. There are

three reasons for this change. The first reason is that in many cases it is hard to find
a variable that is specific to a nest instead of an alternative. So, one often ends up
throwing nonsense variables into the specification of nest-specific explanatory variables
and constraining their coefficients to zero. The second reason is that for the RUMNL
model, it does not make a difference at all if a nest-specific variable is specified for a
nest or for all alternatives within the nest. The third reason is that it greatly simplifies
the syntax and makes it equivalent to the syntax of `clogit` except for the additional
options.

The option `d1` of `nlogit` does not exist for `nlogitrum`. The current version uses the
`ml` method `d0`.

6.4 Predictions

The syntax for `predict` after `nlogitrum` is nearly identical to the syntax after `nlogit`
estimation. The only difference is that the options `xbb` and `xbb#` are replaced by the
option `xb`, since the linear prediction can only be sensibly defined for the bottom level
(the alternatives).

6.5 Options

`group(`*varname*`)` is not optional; it specifies the identifier variable for the groups.

`nests(`*altsetvarB*$\big[$. . . *altsetvar2 altsetvar1*$\big]$`)` is not optional; it specifies the nesting
  structure.

`notree` specifies that the tree structure of the nested logit model is not to be displayed.

`nolabel` causes the numeric codes rather than the label values to be displayed in the
  tree structure of the nested logit model.

`clogit` specifies that the initial values obtained from `clogit` are to be displayed.

`level(`#`)` specifies the confidence level, in percent, for confidence intervals of the coef-
  ficients; see [R] **level**.

`nolog` suppresses the iteration log.

`robust` specifies that the Huber/White/sandwich estimator of variance is to be used
  in place of the traditional calculation; see [U] **23.11 Obtaining robust variance
  estimates**.

`ivconstraints(`*string*`)` specifies the linear constraints of the inclusive value parameters.
  One can constrain inclusive value parameters to be equal to each other, equal to fixed
  values, etc. Inclusive value parameters are referred to by the corresponding level
  labels; for instance, `ivconstraints(fast = family)` or `ivconstraints(fast=1)`.

constraints(*numlist*) specifies the linear constraints to be applied during estimation. Constraints are defined using the `constraint` command and are numbered; see [R] **constraint**. The default is to perform unconstrained estimation.

maximize_options control the maximization process; see [R] **maximize**. You will likely never need to specify any of the *maximize_options*, except for `iterate(0)` and possibly `difficult`. If the iteration log shows many "not concave" messages and is taking many iterations to converge, you may want to use the `difficult` option to help it converge in fewer steps.

6.6   Options for predict

Consider a nested logit model with three levels: $P(ijk) = P(k|ij) * P(j|i) * P(i)$.

pb, the default, calculates the probability of choosing bottom-level alternatives,
   `pb = P(ijk)`.

p1 calculates the probability of choosing first-level alternatives, `p1 = P(i)`.

p2 calculates the probability of choosing second-level alternatives,
   `p2 = P(ij) = P(j|i)*P(i)`.

xbb calculates the linear prediction for the bottom-level alternatives.

xb1 calculates the linear prediction for the first-level alternatives.

xb2 calculates the linear prediction for the second-level alternatives.

condpb, `condpb = P(k|ij)`.

condp1, `condp1 = P(i)`.

condp2, `condp2 = P(j|i)`.

ivb calculates the inclusive value for the bottom-level alternatives:
   `ivb = ln(sum(exp(xbb)))`, where `xbb` is the linear prediction for the bottom-level alternatives.

iv2 calculates the inclusive value for the second-level alternatives:
   `iv2 = ln(sum(exp(xb2 + tau_j*ivb)))`, where `xb2` is the linear prediction for the second-level alternatives, `ivb` is the inclusive value for the bottom-level alternatives, and `tau_j` are the parameters for the inclusive value.

6.7   Generating dummy nests: nlogitdn

The command `nlogitdn` is a wrapper for `nlogit`. Its syntax is equivalent to the `nlogit` syntax. `nlogitdn` analyzes the specified nesting structure, adds appropriate dummy nests and constraints to the specification as discussed in Section 5.3, and calls `nlogit`. It was used for the estimation of model N in Table 6.

6.8   Examples

In order to help the reader become accustomed to the syntax, the commands used
to produce the example models A through N are listed below. Most variable names
should be self-explanatory. The variable `grp` identifies the observations, and the variable
`travel` identifies the alternatives and takes the values 0 for air, 1 for train, 2 for bus,
and 3 for car. The variable `mode` is the 0/1 coded dependent variable. For most NMNL
models, the nesting structure is depicted in Figure 1. The respective variable `type` was
generated using `nlogitgen`. For the models I through L, the nesting structure according
to Figure 2 was generated with the variable `typedeg`:

```
. nlogitgen type = travel(public: 1 | 2, other: 0 | 3 )
new variable type is generated with 2 groups
lb_type:
            1 public
            2 other
. nlogitgen typedeg = travel(public: 1 | 2, air: 0, car: 3)
new variable typedeg is generated with 3 groups
lb_typedeg:
            1 public
            2 air
            3 car
```

Since no variables enter the models on the level of the nests, the nonsense variables
`nothing1` and `nothing2` were generated. The constraints that show up in the `nlogit`
commands constrain their coefficients to zero. The models themselves were estimated
using the following commands:

```
. * Model A:
. clogit mode asc_* hinc_* time_*, group(grp)

. * Model B:
. clogit mode asc_* hinc_* time time_air, group(grp)

. * Model C:
. nlogitrum mode asc_* hinc_* time_*, group(grp) nests(travel type)

. * Model D:

. * Model E:
. nlogit mode (travel = asc_* hinc_* time_* )(type=nothing1), group(grp) const(
> 1) d1

. *Model F:
. nlogit mode (travel = asc_* hinc_* time time_air )(type=nothing1), group(grp)
>  const(1)

. * Model G:
. nlogit mode (travel = asc_* hinc_* time time_air)(type=nothing1), group(grp)
> const(1) ivc(other=public)

. * Model H:
. nlogitrum mode asc_* hinc_* time time_air, group(grp) nests(travel type) ivc(
> other=public)

. * Model I:
. nlogitrum mode asc_* hinc_* time , group(grp) nests(travel typedeg) ivc(air=3
> .14159, car=3.14159)

. * Model J:
```

```
. nlogit mode (travel = asc_* hinc_* time )(typedeg=nothing2), group(grp) const
> (2)
. * Model K:
. nlogitrum mode asc_* hinc_* timepublic time_air time_car, group(grp) nests(tr
> avel typedeg) ivc(air=3.14159, car=3.14159)
. * Model L:
. nlogit mode (travel = asc_* hinc_* timepublic time_air time_car)(typedeg=noth
> ing2), group(grp) const(2) ivc(air=1, car=1)
. * Model M = Model E
. * Model N:
. nlogitdn mode (travel = asc_* hinc_* time time_air)(type=nothing1), group(grp
> ) const(1)
```

Note that the IV parameters of `air` and `car` in models I and K do not actually exist as discussed in Section 5.2. Since the algorithm does not realize this beforehand, these parameters have to be restricted to an arbitrary nonzero number (in the examples, 3.14159 was chosen to illustrate the arbitrariness).

7 Conclusions

The name "nested logit" has been given to different models. This paper argues and demonstrates that the seemingly slight difference in the specification of the outcome probabilities can lead to substantially different results and interpretations thereof. Researchers using a nested logit model (and the readers of their results) should be aware of the actual variant used.

One of these variants (called RUMNL in this paper) is derived from a random utility maximization (RUM) model that is prevalent in econometrics. The estimated coefficients can be readily interpreted and simple tests like asymptotic t tests directly test hypotheses of interest. This holds irrespective of the type of included explanatory variables and specified nesting structure.

The alternative (called NNNL in this paper) implies a varying scaling of the underlying utilities across alternatives. Depending on the model specification, it can give equivalent results to those of RUMNL, and the structural parameters can be recovered. But in order to do so, the estimated coefficients have to be rescaled, and this also has to be kept in mind for hypothesis tests. This is the case if only alternative-specific parameters enter the model. If generic variables (variables with a common coefficient across alternatives) are present, the NNNL model places restrictions on the parameters that are often counterintuitive and undesired. The reason is that the inclusive value parameters in this case not only constitute the (dis)similarities of the alternatives, but also the different scaling of the generic variable coefficients across nests.

Stata 7.0 comes with an implementation of the NNNL model. This paper introduces the Stata package `nlogitrum`, which implements the preferred RUMNL model.

8   Acknowledgments

9   References

Börsch-Supan, A. 1990. On the compatibility of nested logit models with utility maximization. *Journal of Econometrics* 43: 373–388.

Greene, W. H. 2000. *Econometric Analysis.* 4th ed. London: Prentice–Hall.

Hensher, D. A. and W. H. Greene. 2002. Specification and estimation of the nested logit model: alternative normalizations. *Transportion Research* 36(1): 1–17.

Hunt, G. 2000. Alternative nested logit model structures and the special case of partial degeneracy. *Journal of Regional Science* 40: 89–113.

Koppelman, F. S. and C. H. Wen. 1998. Alternative nested logit models: structure, properties and estimation. *Transportation Research-B* 289–298.

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

Maddala, G. S. 1983. *Limited-Dependent and Qualitative Variables in Econometrics.* Cambridge, MA: Cambridge University Press.

McFadden, D. 1974. Conditional logit analysis of qualitative choice behavior. In *Frontiers in Econometrics*, ed. P. Zerembka, 105–142. New York: Academic Press.

—. 1981. Econometric models of probabilistic choice. In *Structural analysis of discrete data and econometric applications*, eds. C. F. Manski and D. L. McFadden, 198–272. Cambridge, MA: MIT Press.

Train, K. 2002. *Discrete Choice Methods with Simulation.* Cambridge University Press.

About the Author

Correspondence to: Florian Heiss; University of Mannheim; MEA; L13,17; 68131 Mannheim; Germany. Phone: +49-621-181-1858. Fax: +49-621-181-1863.
Email: heiss@econ.uni-mannheim.de