Journal of Choice Modelling 26 (2018) 28–40

Contents lists available at ScienceDirect

Journal of Choice Modelling

journal homepage: www.elsevier.com/locate/jocm

Computational methods for estimating multinomial, nested, and cross-nested logit models that account for semi-aggregate data

Jeffrey P. Newman [a], Virginie Lurkin [b], Laurie A. Garrow [c,*]

[a] Cambridge Systematics, Inc., 115 South LaSalle Street, Suite 2200, Chicago, IL, 60603, USA
[b] École Polytechnique Fédérale de Lausanne, Route Cantonale, 1015, Lausanne, Switzerland
[c] Georgia Institute of Technology, 790 Atlantic Drive, Atlanta, GA, 30332-0335, USA

ARTICLE INFO

Keywords:
Discrete choice models
Semi-aggregate data
Airline itinerary choice models

ABSTRACT

We present a summary of important computational issues and opportunities that arise from the use of semi-aggregate data (where the explanatory data for choice scenarios are not necessarily unique for each decision-maker) in discrete choice models. These data are encountered with large transactional databases that have limited consumer information, a common feature in some transportation planning applications, such as airline itinerary choice modeling. We developed a freeware software package called Larch, written in Python and C++, to take advantage of these kind of data to greatly speed the estimation of discrete choice model parameters. Benchmarking experiments against Stata (a commonly used commercial package), Biogeme (a commonly used freeware package), and ALOGIT (a highly specialized commercial package for discrete choice modeling) based on an industry dataset for airline itinerary choice modeling applications shows that the size of the input estimation files are 50–100 times larger in Stata and Biogeme, respectively. Estimation times are also much faster in ALOGIT and Larch; *e.g.*, for a small itinerary choice problem, a multinomial logit model estimated in ALOGIT or Larch converged in less than one second whereas the same model took almost 15 seconds in Stata and more than three minutes in Biogeme.

1. Introduction

Discrete choice models are the backbone of empirical analysis in many fields, including transportation, economics, marketing, public policy and operations research. As its name suggests, "discrete choice models" are used to model how individuals select one (or in some cases more than one) discrete alternative from a set of mutually exclusive and collectively exhaustive alternatives (Ben-Akiva and Lerman, 1985; Train, 2003). An early mathematical form of a discrete choice model was introduced by Daniel McFadden, who in 1972 used a multinomial logit (MNL) model to forecast ridership for the Bay Area Rapid Transit (BART) system (McFadden, 2001). The MNL model is still widely used in various applications because of its mathematical elegance and simplicity, although it is often criticized for employing unrealistic assumptions about behavior. Over the years since the development of the MNL, researchers have derived and estimated dozens of other discrete choice models that have relaxed one or more restrictions associated with the MNL model. For example, the nested logit (NL) (McFadden, 1978; Williams, 1977) and cross-nested logit (CNL) (Vovsha, 1997) incorporate more realistic substitution patterns by relaxing the assumption that error terms are independent across

* Corresponding author.
E-mail addresses: jnewman@camsys.com (J.P. Newman), virginie.lurkin@epfl.ch (V. Lurkin), laurie.garrow@ce.gatech.edu (L.A. Garrow).

https://doi.org/10.1016/j.jocm.2017.11.001
Received 1 February 2017; Received in revised form 21 September 2017; Accepted 1 November 2017

alternatives. The probit (Daganzo, 1979) and mixed logit models (Train, 2003) incorporate random taste variation and can be used for applications in which error terms are correlated across observations.

The objective of this paper is to describe some important computational methods that are especially useful for estimating parameters for choice models with semi-aggregate data (in which the choice scenarios are not necessarily unique across decision-makers). This data feature is commonly encountered with large transactional databases that have limited consumer information. For example, many online retailers have information about the products offered for sale at a given point in time and which of these products were purchased, but do not have information about the consumer who purchased a product. This results in semi-aggregate data in the sense we can aggregate choices for a particular choice set (since we do not have customer-level information). The methods we will describe have been implemented in Larch, a free open-source software package written in Python and C++ that estimates MNL, NL, and CNL models. Larch helps address an emerging problem that many researchers working with large transactional databases face by leveraging more efficient data storage and computational procedures to reduce estimation times for large datasets.

The remainder of this paper describes in detail a number of factors that are relevant for the estimation of discrete choice models for semi-aggregate data. The next section provides an overview of the data (and modeling problem) that motivated the development of Larch. Then we present an overview of some of the discrete choice models that can be estimated efficiently using Larch. The next sections review common data formats (including the identification of chosen and available alternatives), present one method that can be used to number alternatives for large datasets when estimating discrete choice models that contain nests, and describe the methodology used by Larch to weight the log-likelihood function. Finally, we present results from computational experiments based on an industry dataset for airline itinerary choice modeling applications.

2. Problem motivation and data

Itinerary choice models are used to predict the probability that a passenger purchases a specific airline itinerary. Itinerary choice models are used by airlines, aircraft manufacturers, government agencies, etc. and support long-term decisions including where and when to schedule flight legs, how many aircraft to purchase (or manufacture), and which airlines are potentially good code share partners. In itinerary choice models, it is common to construct the set of available itineraries from leg schedule files and/or from revealed purchase transaction data. In the first case, different rules are used to determine which legs can be combined to form multi-leg itineraries. For example, these rules determine minimum and maximum connection times and determine whether legs can be operated by different carriers. In the second case, actual ticket purchases over a longer period (typically a month) are used to construct the set of available alternatives. Although this is not ideal from a theoretical perspective (as itineraries that are infrequently chosen may not be included, potentially leading to bias in parameter estimates), the sheer volume of the ticket transactions helps mitigate this concern (that is, the larger the transaction database, the more likely infrequently chosen itineraries will appear).

The data used for this study were derived from a ticketing database provided by the Airlines Reporting Corporation. The data represent ten origin destination pairs for travel in U.S. continental markets in May of 2013. Itinerary characteristics have been masked, *e.g.*, carriers are labeled generically as "carrier X" and departure times have been aggregated into categories (morning, afternoon, evening). An average fare is provided but is not accurate (a random error has been added to each fare). These modifications were made to satisfy nondisclosure agreements, so that the data used in this paper can be published for teaching and demonstration purposes. It is generally representative of real itinerary choice data used in practice, and the results obtained from this data are intuitive from a behavioral perspective, but it is not accurate and should not be used for behavioral studies.

There are three characteristics of itinerary choice models that heavily influenced our decision to develop Larch. These characteristics include: (1) the number of alternatives in a choice set, (2) the inclusion of different markets in the estimation dataset, and (3) the semi-aggregate nature of the data, in which the choice scenarios are not necessarily unique across decision-makers. With regard to the first point, the number of alternatives in itinerary choice models is large, *e.g.*, Coldren (2005) reports hundreds of itineraries for the estimation dataset used by United Airlines. The maximum number of alternatives we have in our representative dataset is similarly large, as we include as many as 127 alternatives. Although this may not be an especially large choice set compared to certain other contexts (*e.g.*, in residential location choice, there may be thousands of individual alternatives modeled), when considered jointly with the number of observations found in transactional databases (millions, or more), even a few hundred alternatives can be computationally challenging to process. Second, with regards to market segmentation, it is common in itinerary choice models to define a segment as all origin-destination (OD) pairs that share one or more common characteristics. For example, in our earlier work (Lurkin et al., 2017) we defined market segments as a function of distance, direction of travel, and the number of time zones traveled. The inclusion of multiple OD pairs in the same estimation dataset causes some challenges on how to identify nests that share the same characteristics. For example, "alternative 10" in the first OD pair may be operated by American Airlines whereas "alternative 10" in the second OD pair may be operated by United Airlines. Carefully numbering alternatives (described in Section 5) allows us to associate the same set of alternative numbers with itinerary characteristics, but further explodes the number of virtual alternatives in the nest. Finally, because we have no socio-demographic information and because the set of available choices does not vary across individuals, it is very common in our database to have a situation where many individuals face a completely identical choice set, and where some individuals chose the first alternative, some the second alternative, etc. Many existing software applications require that the choice set be "repeated" for each chosen alternative, *e.g.*, that one choice set be defined for those individuals selecting the first alternative, a second (and identical) choice set be defined that for those individuals selecting the second alternative, etc.

Larch was designed to address these data characteristics by storing data in an IDCASE-IDALT format[1] (which helps reduce the size of the input dataset in terms of the number of columns) and appropriately defining a weighting function to account for the semi-aggregate nature of the data (which helps reduce the size of the input dataset in terms of the number of rows). Given an understanding of the data features (that are commonly encountered with large transactional databases that have limited customer information) that motivated the development of Larch, the next section reviews the types of discrete choice models that Larch can estimate.

3. Review of discrete choice models[2]

Discrete choice models are used to predict the probability that an individual selects a discrete alternative among a set of mutually exclusive and collective exhaustive discrete choice alternatives (commonly referred to as the "choice set"). Most statistical models of discrete choice used in practical applications are based on the concept of utility. Utility is a scalar index of value that is a function of attributes and/or individual characteristics. Utility represents the "value" an individual places on different attributes and captures how individuals make trade-offs among different attributes. Individuals are assumed to select the alternative that has the maximum utility. Alternative i is chosen if the utility individual h obtains from alternative i, U_{hi}, is greater than the utility for all other alternatives. Formally, alternative i is chosen *iff* $U_{hi} > U_{hj} \quad \forall j \neq i$. The utility for alternative i and individual h, U_{hi}, has an observed component, V_{hi}, and an unobserved component, commonly referred to as an "error term", ε_{hi}, but is more precisely referred to as the "stochastic term". The utility U_{hi} for individual h in choosing alternative i from choice set $\mathscr{J}_h$ is a linear-in-parameters function of $\mathbf{x}_{hi}$, $U_{hi} = V_{hi} + \varepsilon_{hi} = \boldsymbol{\beta}_i' \mathbf{x}_{hi} + \varepsilon_{hi}$, where U_{hi} is the true utility, V_{hi} is the estimated utility, $\mathbf{x}_{hi}$ comprises attributes of the alternative i and/or decision-maker h and $\boldsymbol{\beta}$ is a vector of estimated coefficients. Various different assumptions about the distribution of ε_{hi} give rise to different choice models.

3.1. Multinomial logit model

If ε_{hi} is distributed independently and identically (*iid*) with a Gumbel (or extreme value type I) distribution, the discrete choice model becomes a multinomial logit (MNL) model (McFadden, 1974) and the probability of individual h choosing alternative i is given as:

$$P(y_h = i|\mathbf{x}_{hi}) = \frac{e^{V_{hi}}}{\sum_{j \in \mathscr{J}_h} e^{V_{hj}}} = \frac{e^{\boldsymbol{\beta}_i' \mathbf{x}_{hi}}}{\sum_{j \in \mathscr{J}_h} e^{\boldsymbol{\beta}_i' \mathbf{x}_{hj}}} \tag{1}$$

where y_h is the outcome variable corresponding to the alternative chosen by individual h. In the MNL model, the assumption that the error terms are *iid* Gumbel is advantageous in the sense that the choice probability takes on a closed-form expression that is computationally simple. However, the same assumption leads to the independence of irrelevant alternatives (IIA), a property which states that the ratio of choice probabilities P_{hi}/P_{hj} for all $i, j \in \mathscr{J}_h$ is independent of the attributes of any other alternative. In terms of substitution patterns, this means a change or improvement in the utility of one alternative will draw market share (in a disaggregate fashion) proportionately from all other alternatives. In many applications, the IIA property may not be realistic. Other models that belong to the Generalized Extreme Value (GEV) class relax the independence assumption by including covariance terms that are created through allocating an alternative to one or more nests while maintaining the assumption that total variance is identically distributed across alternatives.

3.2. Two-level nested logit model

The nested logit (NL) model (Williams, 1977; McFadden, 1978), relaxes the assumption that errors are independently distributed by grouping alternatives into M nests, *i.e.*, $i \in \mathscr{A}_m$, $m = 1, \ldots, M$. Each alternative belongs to one and only one nest. The NL utility function can be expressed as follows (suppressing the index for individual h for notational convenience):

$$U_{im} = V_i + \varepsilon_m + \varepsilon_i \tag{2}$$

That is, the total variance associated with each alternative in nest m is decomposed into a common error component, ε_m, and an independent error term, ε_i. Alternatives that belong to the same nest share a common error term. Instead of assuming that all error terms are independent, we assume that the total variance associated with each alternative, given as $(\varepsilon_i + \varepsilon_m)$ must be identically distributed and follow a Gumbel with mode zero and a scale of one. In addition, we assume that the independent error terms also follow a Gumbel with mode zero, but with a different scale. An intuitive expression for the NL choice probability can be derived as the product of a conditional and marginal probability (this derivation is provided in Train (2003), p. 90). This formulation is particularly helpful when extending NL models to include additional levels of nests.

[1] Larch can read in either the idcase or idcase-idalt format and can perform the data conversions between these two different data structures.
[2] This section draws heavily on the description found in (Garrow, 2010).

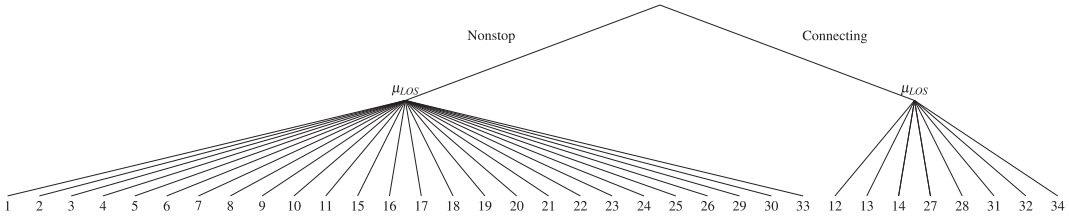

Fig. 1. 2-level NL model, nested by level-of-service.

$$P_i = P_{i|m} \times P_m = \frac{\exp(V_i/\mu_m)}{\sum_{j \in \mathscr{A}_m} \exp(V_j/\mu_m)} \times \frac{\exp(V_m + \mu_m \Gamma_m)}{\sum_{l=1}^{M} \exp(V_l + \mu_l \Gamma_l)},$$
$$\Gamma_m = \log\left(\sum_{j \in \mathscr{A}_m} \exp(V_j/\mu_m)\right), 0 < \mu_m \leq 1 \tag{3}$$

The first component of the product is the probability of selecting alternative i among all j alternatives in nest m, conditional on the choice of m, and the second product is the probability of selecting nest m among all nests. Γ_m is often called the "log-sum term" because it is the log of a sum (this terminology should not to be confused with μ_m, the "logsum parameter"). The logsum parameter, μ_m, is a measure of the degree of correlation and substitution among alternatives in nest m. More precisely, the correlation is given by $1-\mu_m^2$ (Abbe et al., 2007). Higher values of μ_m imply less, and lower values imply more, correlation among alternatives in the nest. In turn, higher correlation leads to greater competition effects among alternatives in the nest.

Fig. 1 shows a two-level NL model that groups alternatives into nests that correspond to level of service, *i.e.*, the first nests contains 26 nonstop itineraries and the second nest contains eight connecting itineraries. In this example, the two-level NL model is used to incorporate increased substitution (controlled by the parameter μ_{LOS}) for alternatives that share the same level of service.[3] Similarly, other two-level NL models could be used to incorporate increased substitution patterns along other dimensions but only for one product dimension at a time (*e.g.*, either carrier or level of service, but not both carrier and level of service).[4]

3.3. Three-level nested logit (NL) model

To incorporate increased substitution across multiple dimensions, the analyst has several options. The first option is to use a NL model that contains three (or potentially more) levels, such as the one shown in Fig. 2. In this example, the upper level nests represent time of day nests, grouping together morning, afternoon, and evening departures (μ_{TOD}), and the lower level nests represent level of service nests for nonstop and connecting flights (μ_{LOS}).[5] The probability of choosing alternative i in a three-level nest is given as:

$$P_i = P_{i|nm} \times P_{nm|m} \times P_m = \frac{\exp(V_i/\mu_{nm})}{\sum_{j \in B_n} \exp(V_j/\mu_{nm})} \times \frac{\exp(\mu_{nm}\Gamma_{\mathrm{nm}}/\mu_m)}{\sum_{k \in A_m} \exp(\mu_{km}\Gamma_{\mathrm{km}}/\mu_m)} \times \frac{\exp(\mu_m \Gamma_m)}{\sum_{l=1}^{M} \exp(\mu_l \Gamma_l)}, \tag{4}$$
$$\Gamma_{nm} = \ln\left(\sum_{j \in B_n} \exp(V_j/\mu_{nm})\right), \Gamma_m = \ln\left(\sum_{k \in Am} \exp(\mu_{km}\Gamma_{km}/\mu_m)\right), 0 \leq \mu_{mn} \leq \mu_m \leq 1$$

The first component of the product is the probability of selecting alternative i among all j alternatives in nest nm, conditional on the choice of nm. The second product is the probability of selecting nest nm among all two-level nests in nest m, conditional on the choice of m. The third product is the probability of selecting nest m among all three-level nests.

To ensure that the three-level NL model is consistent with utility maximization, the correlation must increase as one moves down the tree. In Fig. 2, this implies that the correlation (or substitution among alternatives) for alternative 1 is the strongest (or greatest) with the other morning nonstops (alternatives 4, 7, 8, 17, 18, 19, and 20) and stronger with morning connections (alternatives 13, 27, and 31) than with all other alternatives.

[3] The two-level NL model shown in Fig. 1 constrains the logsum coefficients to be the same across all nests, but this is not a requirement for the two-level NL model.

[4] If the product dimension is defined to include both carrier and level of service, then competition effects for the joint effect of carrier and level of service can be included in a two-level NL model. However, this structure does not provide insight into which of the two dimensions competes more strongly with each other (*e.g.*, do nonstop flights compete more strongly with other nonstop flights or do flights on American Airlines compete more strongly with other American Airlines flights?). To determine the effect of each of these product dimensions, a three-level NL model, described in the next section, can be used.

[5] The three-level NL model shown in Fig. 2 constrains the logsum coefficients to be the same across all nests at the same level, but this is not a requirement for the three-level NL model.

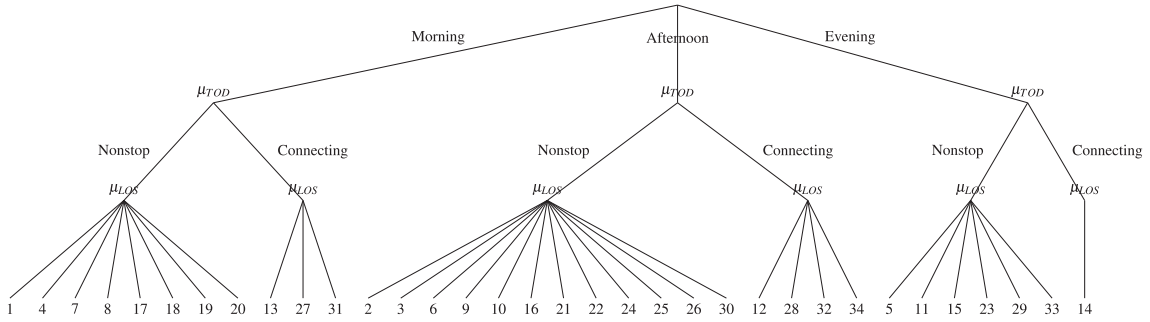

Fig. 2. 3-level NL model, nested by time of day and level of service.

3.4. Ordered generalized extreme value (OGEV) models

To incorporate increased substitution across multiple dimensions, the analyst can also use a discrete choice model that allocates alternatives to more than one nest. One such model is the ordered generalized extreme value (OGEV) model (Small, 1987), used in applications in which the ordering of alternatives has a physical meaning. For example, the OGEV model can be used to capture time of day competition effects among airline itineraries.

The OGEV probability is given as:

$$P_i = \sum_{m=i}^{i+T} P_{i|m} \times P_m = \sum_{m=i}^{i+T} \left[\frac{(\tau_{m-i}\exp(V_i))^{\frac{1}{\mu}}}{\sum_{j\in\mathscr{A}_m} (\tau_{m-j}\exp(V_j))^{\frac{1}{\mu}}} \times \frac{\left(\sum_{j\in\mathscr{A}_m} \tau_{m-j}\exp(V_j/\mu)\right)^{\mu}}{\sum_{r=1}^{J+T} \left(\sum_{s\in\mathscr{A}_r} \tau_{r-s}\exp(V_s/\mu)\right)^{\mu}} \right], \quad 0<\mu\leq 1, \sum_{m=1}^{J+T} \tau_m = 1 \tag{5}$$

where

- T is the number of adjacent time periods connected in the OGEV model,
- J is the total number of time periods,
- $\mathscr{A}_m$ is the set of all alternatives that belong to nest m,
- r is an index used to sum over all nests,
- τ_{m-1} are unknown allocation parameters that characterize the portion of alternative i assigned to a nest. Allocation parameters are non-negative, *i.e.*, $\tau_{m-1} \geq 0$ and must sum to one for every alternative. Defining nests for a T-step OGEV model as shown in Fig. 3, alternative i belongs to nests $i-1, i, \ i+1, \ldots, i+T$, this last condition is equivalent to $\sum_{m=1}^{J+T} \tau_m = 1$
- μ is the logsum coefficient associated with each nest.

The first component of the product is the probability of selecting alternative i among all alternatives that belong to nest m, conditional on choosing nest m. The second product is the probability of selecting nest m among all nests. The total probability for alternative i is obtained by summing over all nests that contain alternative i. The portion of an alternative that shares a nest with itineraries that depart in the time period immediately before (τ_0) or the time period immediate after (τ_1) is estimated from the data. A constraint is also added to ensure that $\tau_0 + \tau_1 = 1$. From an interpretation perspective, a value of $0.5 < \tau_0 < 1$ (and $0 < \tau_1 < 0.5$) means that an itinerary departing in time period three would compete more with itineraries in the earlier time period two than with itineraries in the later time period four. Intuitively, this result would be expected for outbound itineraries for travelers that need to arrive at their destinations by a fixed time.

Fig. 3 shows an OGEV model for the same example choice set as before and with one adjacent time period modeled as competitive (T = 1).[6]

Note that the MNL proportional substitution property applies to those alternatives that do not share a nest in common and for which their covariance term is zero. This may be problematic if there is increased substitution among alternatives separated by more than one time step. This can be accommodated using an OGEV model that allocates alternatives to more than one adjacent time period.

The OGEV model described above can be extended to more than one adjacent time period as shown in Fig. 4. The OGEV model shown in Fig. 4 exhibits greater-than-MNL competition for itineraries that depart in the two time periods immediately before or immediately after. Further, itineraries one adjacent period away compete more than itineraries two adjacent periods away.

In practice, the number of time-periods (or T) is determined by comparing model fits between a model that incorporates t adjacent

[6] The OGEV models shown in Fig. 3 constrain the logsum coefficients to be the same across all nests, but this is not a requirement for the OGEV model.

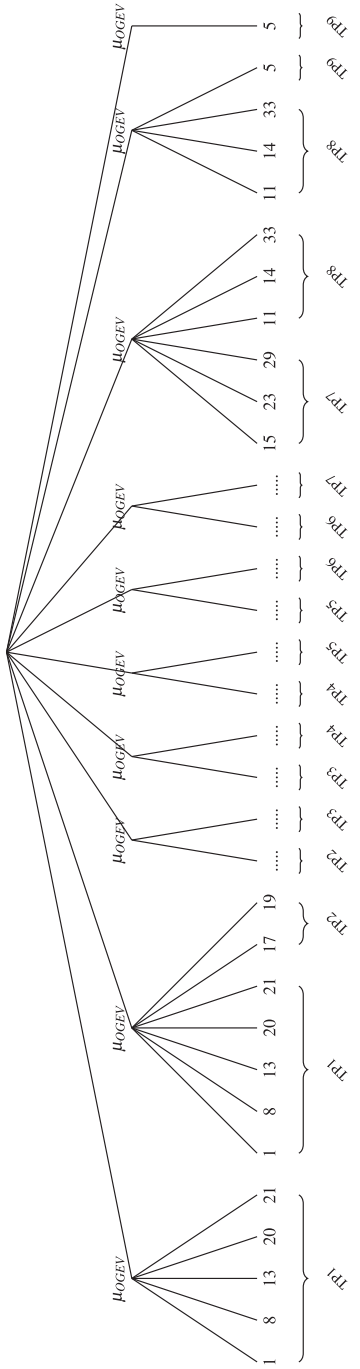

Fig. 3. Constrained OGEV model with one adjacent time period (T = 1).

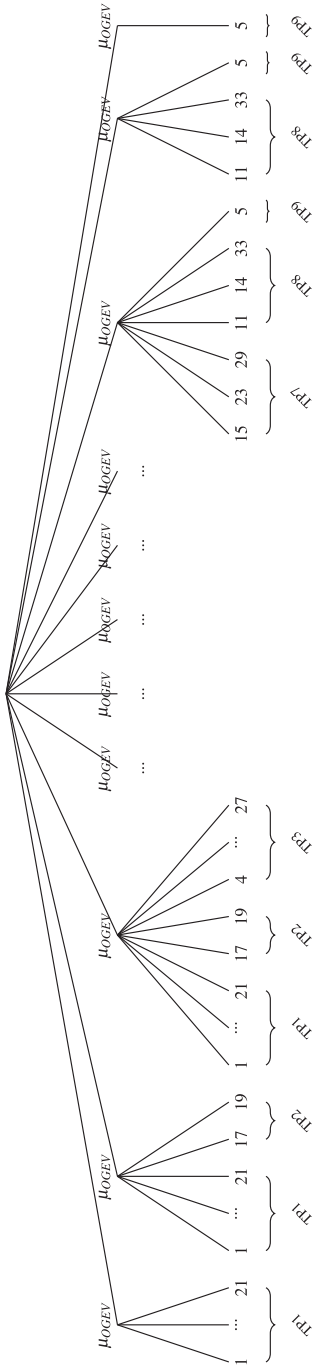

Fig. 4. Constrained OGEV model with two adjacent time periods (T = 2).

Table 1
Data in IDCASE-IDALT format.

(a) Original Data

ID_CASE	ID_ALT	CHOICE	COST	N_CNXS
1	1	1	450	0
1	2	0	350	1
1	3	0	300	2
2	1	0	450	0
2	2	0	350	1
2	3	1	300	2
3	1	0	450	0
3	2	1	350	1
3	3	0	300	2
4	1	0	350	1
4	2	1	300	2
5	1	0	350	1
5	2	1	300	2

(b) Weighted Data

ID_CASE	ID_ALT	CHOICE	WEIGHT	COST	N_CNXS
1	1	1	**1**	450	0
1	2	0	**1**	350	1
1	3	0	**1**	300	2
2	1	0	**1**	450	0
2	2	0	**1**	350	1
2	3	1	**1**	300	2
3	1	0	**1**	450	0
3	2	1	**1**	350	1
3	3	0	**1**	300	2
–	–	–	–	–	–
–	–	–	–	–	–
4	1	0	**2**	350	1
4	2	1	**2**	300	2

Note: Blank lines are retained in the graphical representation here to clearly show changes, but are actually completely removed in application.

time periods with a model that incorporates $t+1$ adjacent time periods, *e.g.*, the analyst compares the model fits between a OGEV model with $t = 1$, $t = 2$, $t = 3$, etc. Sufficient time periods have been incorporated when there is "little" improvement observed in the fit between two models. Statistical tests can be used to determine if the log likelihood values for the two models are statistically equivalent.

3.5. Cross-nested logit models

The OGEV models described above are a special case of the cross-nested logit (CNL) model, which has also been called a -generalized nested logit (GNL) model in the literature (Vovsha, 1997; Wen and Koppelman, 2001). Unlike the OGEV model, the CNL

Table 2
Data in IDCASE format.

(a) Original Data

ID_CASE	CHOICE	COST1	N_CNXS1	COST2	N_CNXS2	COST3	N_CNXS3
1	1	450	0	350	1	300	2
2	3	450	0	350	1	300	2
3	2	450	0	350	1	300	2
4	2	350	1	300	2	−1	−1
5	2	350	1	300	2	−1	−1

(b) Weighted Data

ID_CASE	CHOICE	WEIGHT	COST1	N_CNXS1	COST2	N_CNXS2	COST3	N_CNXS3
1	1	**1**	450	0	350	1	300	2
2	3	**1**	450	0	350	1	300	2
3	2	**1**	450	0	350	1	300	2
–	–	–	–	–	–	–	–	–
4	2	**2**	350	1	300	2	−1	−1

Note: Blank lines are retained in the graphical representation here to clearly show changes, but are actually completely removed in application.

Table 3
Larch IDCASE-IDALT format.

ID_CASE	ID_ALT	CHOICE	COST	N_CNXS
1	1	**1**	450	0
1	2	**1**	350	1
1	3	**1**	300	2
2	1	**0**	350	1
2	2	**2**	300	2

is more "general" in the sense that its nesting structures are not restrictive and both allocation and logsum parameters can be estimated (although in practice restrictions on these parameters are often imposed to facilitate estimation and model interpretation). Like the OGEV models, the CNL is based on a two-level nesting structure. In Fosgerau et al. (2013), the authors have shown that any choice model can be approximated as precisely as desired by a CNL model. The probability equation for the CNL model is shown below. Correlation among alternatives is now a function of both the τ allocation weights and the μ logsum coefficients.

$$P_i = \sum_m \left[\frac{(\tau_{im} \exp(V_i))^{\frac{1}{\mu_m}}}{\sum_{j \in \mathscr{N}_m} (\tau_{jm} \exp(V_j))^{\frac{1}{\mu_m}}} \times \frac{\left(\sum_{j \in \mathscr{N}_m} \tau_{jm} \exp(V_j/\mu_m) \right)^{\mu_m}}{\sum_l \left(\sum_{j \in \mathscr{N}_l} \tau_{jl} \exp(V_s/\mu_l) \right)^{\mu_l}} \right], \tag{6}$$

$$0 < \mu_m \leq 1, \sum_m \tau_{jm} = 1 \tag{7}$$

4. Data formats

Preparing data for estimation is considered to be a simple process that is often overlooked in the literature, although in any practical application it is an important part of the workflow, and different software tools use data in different fundamental input formats. Commonly used discrete choice modeling packages differ in terms of whether data should be specified in the IDCASE-IDALT format (as required by Stata (Stata Statistical Software: Release 14., 2015), a well known commercial statistical tool), or in the IDCASE format (as required by Biogeme (Bierlaire, 2016), a widely used free open source tool for discrete choice model estimation, and ALOGIT, a specialized commercial tool).

We will use a simplified example to illustrate the different data formats. Suppose we have asked five individuals to choose their preferred itinerary to go from Atlanta to San Francisco. We will assume that only three itineraries are possible:

- A nonstop flight that costs $450,
- A one-stop flight with a connection in Chicago that costs $350, and
- A two-stop flights with connections in San Antonio and Denver that costs $300.

The three first individuals can choose among the three itineraries, whereas the last two individuals can only choose between the two connecting itineraries (*e.g.*, the nonstop has sold out at the time they are shopping).

In the IDCASE-IDALT format required by Stata, depicted in Table 1(a), each row contains information about a single alternative associated with an individual. The ID_CASE column identifies the individual and the ID_ALT column identifies the alternatives available for each individual. The non-availability of one of the itineraries for individuals 4 and 5 is reflected in the data by omitting the row for that alternative. The chosen alternative is identified by a value of one in the CHOICE column, whereas zeros indicate the alternatives that were not chosen. The two last columns, COST and N_CNXS, are the two variables that characterize the different alternatives. The next section describes how alternative IDs can be consistently numbered to create nests.

In the IDCASE format required by Biogeme and ALOGIT and depicted in Table 2(a), each row contains information about all alternatives associated with an individual. The ID_CASE column continues to identify the individual. There is no specific column for the alternatives as all alternatives will be represented in the same row. The CHOICE column identifies the alternative number of the

Table 4
Numbering alternatives.

ID_CASE	ID_ALT	CHOICE	COST	N_CNXS	ID_ALT_LOS
1	1	1	450	0	1
1	2	1	350	1	2
1	3	1	300	2	3
2	**2**	0	350	1	2
2	**3**	2	300	2	3

chosen alternative (1, 2, or 3 in our case). The columns COST and N_CNXS are now specific to each alternatives (there are three columns for COST and three columns for N_CNXS as there are up to three alternatives available for each individual). In this format, the non-availability of an alternative for an individual is represented by the presence of negative values for COST and N_CNXS variables.[7]

All of the software we evaluated allows the use of *weights* in the input data file. Weights are useful when different individuals have chosen the same alternatives among the same set of available alternatives. In our example, for individuals 4 and 5, the explanatory and dependent (choice) variables are all identical for both cases. In such a situation, a WEIGHT column can be added to the input data file and identical cases (the same choice scenarios with same output) can be grouped together. In our example, this is seen by the weight of 2 that has been added for ID_CASE 4 in Table 2(b). Note that the weight of 1 needs to remain for ID_CASEs 1 to 3 for Stata and Biogeme because each individual has chosen a distinct alternative. The two data structures with weights are illustrated in Tables 1(b) and 2(b). Note that all four representations of this dataset are functionally equivalent.

Unlike Stata, Biogeme, or ALOGIT, Larch does not demand the use of one or the other of these data formats. Instead it can use either, or both simultaneously. This allows the analyst to select the preferred data format for each problem. In making that selection, the analyst should consider data storage and computational effort, both of which will vary depending on the details of the desired model and available data.

In addition to allowing the use of either data format, Larch allows cases to be merged not only when they are identical across all explanatory and dependent (choice) data, but also when they are identical across all explanatory data but vary in choices. ALOGIT also allows this, although only for the estimation of MNL models. This semi-aggregate data is observed in the example data above: the first three individuals face the same choice scenarios. Only the outcome of these choice scenarios are different. Larch allows aggregating these individuals into the same case, which leads to an even more compact data format. Table 3 illustrates the resulting input data file for Larch. The CHOICE column indicates the number of individuals who have chosen the corresponding alternative.

This can seem an insignificant detail but, as we will show in below, this feature can have an important impact on the size of the input data file, which in turn can lead to significant differences in the computational resources, and by extension running time, needed to estimate a model.

5. Numbering of alternatives

The simple itinerary choice example shown here has up to three alternatives, but they are not ordered or numbered in a regular way; each elemental alternative has an arbitrary code number assigned to it as a label, and the code numbers for one case are not comparable to another case. For example the ID_ALT 2 in the first case corresponds to the one-stop itinerary, whereas the ID_ALT 2 in the last case corresponds to the two-stop itinerary. These numerical labels are unimportant in the context of an MNL model where all alternatives interact with each other in a homogenous fashion, but they take on an important role for other models where the relationships among alternatives are not homogenous. In those cases, correctly labeling alternatives so as to efficiently associate them with the correct nesting structures is important.

In our simplified example, one can easily imagine jumping directly to a more consistent simple numbering scheme, but it becomes a less trivial matter in a larger context where the itinerary choice set varies much more widely from traveler to traveler, for example when more origin-destination pairs or travel days are considered together. Thus it is helpful to explicitly create an algorithm for numbering alternatives.

As described above, when advanced discrete choice models have to be estimated, alternatives are generally grouped into several nests according to some criteria. Suppose for example that we want to test a two-level nested logit model which has two nests based on the level of service (LOS): nonstop itineraries versus connecting itineraries. To assign each alternative into this structure, we want to be able to easily identify the alternative numbers that correspond to nonstop itineraries and the alternative numbers that correspond to connecting itineraries, for each observation in the data.

To establish a new numbering scheme, we start by defining the relevant categories and their values. For example, for a two-level NL model based on the level of service, two relevant categories can be LOS1 (for nonstop itineraries) and LOS2 (for connecting itineraries). In that scenario, the categorical variable LOS1 takes the value one if the variable N_CNX is equal to zero and the value zero otherwise. Similarly, the categorical variable LOS2 takes the value one if the variable N_CNX is not equal to zero and the value zero otherwise. To number the alternatives in a rational way, we can renumber the alternatives, grouping by the two categorical variables. To ensure the numbering is consistent, the smallest LOS2 alternative number will need to be one larger than the largest LOS1 alternative number. Thus we will need to find the maximum number of LOS1 and LOS2 alternatives in any case across the entire dataset.

For the itinerary choice example of Table 3, there is at most one nonstop (LOS1) itinerary, and at most two connecting (LOS2) itineraries. We therefore reserve the first alternative number (ALT 1) for nonstop itineraries, and the next two numbers (ALT 2 and ALT 3) for connecting itineraries. Within each case, alternatives are re-assigned numbers sequentially within the reserved blocks. The resulting renumbered data is shown in Table 4, with the changes from Table 3 bolded. The example here is simple, but the same logic can be applied to any number of categorical nests (*e.g.*, sufficient sized blocks can be reserved for all itineraries sharing the same level of service, carrier, and time of day).

It is important to note that the numbering scheme used is tied explicitly to the nesting structure implemented. If LOS is not one of the criteria that defines the nesting structure, then it is not necessary to number the alternatives to be able to link the alternative

[7] Note that zeros values could be used but zeros can lead to a misinterpretation regarding the N_CNXS variable.

Table 5
Model estimation running times.

	Stata	Biogeme	ALOGIT	LARCH
Data loading	4.87 s	9 min 10 s	2.4 s	6.63 s
MNL	15.34 s	3 min 9 s	0.11 s	0.72 s
2-level NL	7 min 37 s	286 min 34 s	2.79 s	2.06 s
3-level NL	21 min 41 s	NA	3.51 s	2.30 s
Constrained OGEV with 1 adjacent time period	NA	3 days 23 h	NA	2.14 s
Constrained OGEV with 2 adjacent time periods	NA	4 days 23 h	NA	2.50 s

NA = This software tool is not able to estimate this kind of model without a significant amount of custom coding.

number to the LOS attributes. This can be important for large models, as the number of unique alternative numbers that need to be reserved can be decreased (thus decreasing the memory to estimate the model) by reducing the dimensionality of the relevant categories.

6. Methodology for weighting semi-aggregate data

Traditional disaggregate discrete choice modeling data are often stored based on the assumption that each decision-maker faces a unique choice scenario, where there is some set of alternatives with some set of attributes, and each decision-maker has the opportunity to choose one and only one alternative from that set. In our airline itinerary choice data, this is not exactly correct, because the choice scenarios tend to be repetitive instead of unique. That is, for any discrete choice observation, there are on average about nine other observations that share completely identical choice sets with completely identical choice attributes. This presents an opportunity to group these observations together, to achieve a significant gain in computational speed and a reduction in the overall memory requirements.

For all choice problems, an observation represents a decision-maker, a vector of attributes associated with the decision-maker and alternatives, and the chosen alternative. The problem of interest is to solve for the parameters $\beta^\star$ given a random sample of observations. Estimators based on maximum likelihood estimation are most commonly used to find $\beta^\star$. (Other more complex estimators can be used such as those based on the method of moments or the method of scores (*e.g.*, see Train (2003)), but we focus on maximum likelihood estimators here.) Maximum likelihood estimation solves for the values of β that maximize the likelihood function:

$$L(\beta) = \prod_{h \in \mathscr{H}} \prod_{i \in \mathscr{J}_h} P(i|\mathbf{X}_h, \boldsymbol{\beta})^{d_{hi}} \tag{8}$$

where

- $\mathscr{H}$ is the set of individuals in the random sample,
- $\mathscr{J}_h$ are the available alternatives in the choice set for individual h,
- $\mathbf{X}_h$ is the vector of attributes associated with individual h,
- d_{hi} is an indicator variable equal to 1 if $y_h = i$ (*e.g.*, if individual h selects alternative i), and 0 otherwise,
- $P(i|\mathbf{X}_h, \boldsymbol{\beta})$ is the probability of selecting alternative i given attributes $\mathbf{X}_h$ and estimates $\boldsymbol{\beta}$. (Earlier discussions defined this probability as P_{hi}. When the conditional form is used in this section, it is used to emphasize the fact that the probability is dependent on characteristics of the sampling distribution related to attributes and estimates.)

Computationally, it is easier to maximize the logarithm of the likelihood function, *i.e.*, the log likelihood (LL) function. Moreover, because probabilities must sum to 1, it is typical to actually compute the vector of probabilities $\mathbf{P}(\mathbf{X}_h, \beta)$ that represent the probability of choosing each of the possible alternatives, and then selecting the term $[\mathbf{P}(\mathbf{X}_h, \beta)]_{y_h}$ from this vector for use in computing the log likelihood, or more generally,

$$LL(\beta) = \sum_{h \in \mathscr{H}} \sum_{i \in \mathscr{J}_h} d_{hi} \log([P(i|\mathbf{X}_h, \boldsymbol{\beta})]_i) \tag{9}$$

The structure of Eq. (9) is useful because it highlights an important feature: the value of the vector $\mathbf{P}(\mathbf{X}_h, \beta)$ is not dependent on d_{hi}. Conceptually, the calculation of probability does not depend on the choice. As long as the choice set and the relevant attributes of the choices remain identical across a subset of observations, then most of the computations are common across these observations. Only the relatively inexpensive final computational step explicitly shown in Eq. (9) depends on the choices. Therefore, for any subset of observations in $\mathscr{H}$ where the vector of values for $\mathbf{X}_h$ (and by extension, $P(i|\mathbf{X}_h, \boldsymbol{\beta})$) is identical, the data can be condensed by treating d_{hi} not merely as a binary indicator but as a cardinal variable δ_{hi} (which can take values other than 0 or 1), and summing δ_{hi} over the subset. This leads to a condensed sample of observations $\mathscr{Z}$ derived from $\mathscr{H}$, where every row of $\mathbf{X}_z$ is unique and $\boldsymbol{\delta}_z$ can contain values greater than 1, as well as multiple nonzero values in each record. Since much of the computational effort associated with LL is calculating $\mathbf{P}$, there are significant gains available from combining these cases together, and only calculating the probability once per

row of $\mathscr{Z}$, instead of once per row of $\mathscr{H}$.

This condensed sample is exactly analogous to the semi-aggregate data storage discussed above. For example, when the sample is not condensed, the data might appear as in Table 2(a), and the calculations to generate P are completed once for each case of that table (*i.e.*, five times). When that same data is condensed as (with no loss of information), the calculations to generate P are completed once for each case in Table 3 (*i.e.*, only two times).

Of course, this level of optimization is only possible when using software that treats the choice as a cardinal variable that can take on values other than 0 or 1. When the choice specification is nominal (*i.e.*, given by name or code number, not variable value, as in Biogeme), only one alternative can be designated as chosen, and therefore only a portion of these gains can be achieved by condensing records that are identical across both $\mathbb{X}$ and all zero values in δ. In this case, a separate weighting variable w_z can be introduced, indicating the number of observations condensed in this manner. If the observations are already weighted, the condensed weighting can be calculated simply by summing the relevant weights, instead of counting the observations. Weighted observations are common in statistical applications, so most major software tools for discrete choice modeling are designed to incorporate weighting factors, even if the choice specification must be nominal. This weighted sample approach would use data as is shown in Table 2(b), and the calculations to generate P would be completed once for each case in that table (*i.e.*, four times).

More generally, the information in the statistical weighting factor is interchangeable with the sum of the cardinal choice values, $\sum_{i \in \mathscr{J}_z} \delta_{zi}$. When observation weights are included, they can be expressed purely in the choices by multiplying the observations cardinal choice values by the observations weights, and dividing the weights by a similar amount (essentially, assigning unit weight to all observations). Alternatively, it is possible to linearly rescale the cardinal choices so that the sum of δ across each observation is exactly 1, and the total number of discrete observations (or the total weight of those discrete observations) is expressed as w_z. Either of these methods of using the weights is valid and will give the same result for calculating the model log likelihood or its derivatives. However the second approach (with unit totals of $\boldsymbol{\delta}_z$) is convenient for computing a related measure: the Berndt-Hall-Hall-Hausman (BHHH) matrix (Berndt et al., 1974).

Some optimization algorithms or robust covariance estimators employ the BHHH matrix, which is computed by taking the sum across observations of the outer product of the per-observation gradient (also called the scores), sometimes written as $G^T G$, where G is an array with rows for each observation and columns for each model parameter. When the per-observation gradients are taken from each row of the condensed sample $\mathscr{Z}$, the resulting matrix will be different from the result that would be obtained if the per-observation gradients were taken from $\mathscr{H}$. This is because the weighting (implied by having non-unit values of δ associated with $\mathscr{Z}$) is entering the computed matrix twice, once in G and once in G^T. To reproduce the original BHHH matrix for a discrete choice model using a condensed sample, the weight must enter the computed matrix only once. This is conveniently achieved when $\sum_{i \in \mathscr{J}_z} \delta_{zi} = 1$ for all i, because the impact of the weight is then isolated to w_z, and we can compute the BHHH matrix by $\sum_z w_z G_z^T G_z$ where G_z is the row of G associated with condensed observation z.

7. Implementation comparison

Our example itinerary choice dataset represents 235,198 passengers, making purchases from 4515 itineraries serving 10 city pairs, with each passenger purchasing from one of 105 distinct choice sets (defined by origin, destination, day of week, and travel class). Each choice set contains at least 32 possible itineraries, and up to a maximum of 127 possible itineraries.

We estimated itinerary choice models corresponding to each of the discrete choice model forms described above, and depicted in Figs. 1–4. For each model, we ran the estimation in Stata, Biogeme, ALOGIT (ALOGIT 4.3, 2016), and Larch, to compare the results.[8] All models were run on a Windows Server 2008 R2 with a 64 bit processor and two Xeon 2.40 GHz processors with a total of 16 cores and 288 GB of RAM. The version of Stata we used (in addition to Biogeme and Larch) were able to take advantage of the multi-processors. ALOGIT is not designed for multiple processors, although like Larch, it is designed explicitly for discrete choice modeling and is highly optimized for the estimation of linear-in-parameters MNL models. It employs all of the optimizations outlined in this paper for estimating MNL models, although the NL implementation in ALOGIT does not allow for using the semi-aggregate condensed sample data structure.

The data used in Stata were prepared as described above in IDCASE-IDALT format. This data format required 286,587 rows and 14 columns, resulting in a datafile of approximately 13 MB. The data used in Biogeme were prepared in IDCASE format, with 4515 rows and 1272 columns; this datafile was approximately 26 MB. For Larch, the data was prepared in the condensed format described above, which required only 6023 rows and 13 columns, yielding a total data file size of only 260 KB. For ALOGIT, the data were prepared in a modified (IDCASE style) version of the condensed format described above, which required only 105 rows but 1007 columns, yielding a total data file size of 350 KB. The computational times for estimating each model with our sample dataset are shown in Table 5. In large part due to the computational efficiencies described in the sections above, Larch was able to achieve speeds that are orders of magnitude better than less well optimized tools available tools, especially on more advanced choice models. The converged log likelihoods and estimated parameters for each model were found to be the same for each model estimated across all software tools; to conserve space, and because the models are based on representative but not actual data, the estimation results from these models are not presented in full here.

[8] Stata, Biogeme, and ALOGIT were chosen as representative high quality reference examples, and this experiment is by no means an exhaustive survey of the performance of the various tools that might be used for discrete choice model estimation in general, or itinerary choice models in particular.

These results shown in Table 5 are based on publicly available sample data. These data are limited in size to provide an illustration of model estimation techniques at a comparatively low computational cost. The computational times are not representative of the effort required to estimate models using the much larger datasets of airline itinerary choices available from proprietary sources. For example, we extracted the sample data from a dataset containing over 10 million passengers making choices from over 1 million distinct choice sets. We tested the estimation of a similar MNL model with the complete data using both Larch and ALOGIT. We found that both programs performed similarly well for this large dataset, with Larch completing in 8 min 42 s, and ALOGIT completing in 14 min 23 s.[9]

Summary

This paper presented a discussion of computational issues surrounding the estimation of airline itinerary choice models using large volumes of semi-aggregate transactional data. We showed that substantial gains in data storage and computational efficiency are available when discrete choice modeling tools are carefully constructed to take advantage of the structural features of semi-aggregate data.

Larch is made available for free under the GNUv3 license. Precompiled binaries for macOS and Windows are available though the python package index (PyPI), and the source code is available on GitHub at https://github.com/jpn–/larch. Documentation for Larch is available at https://larch.readthedocs.io, along with sample code and datasets to estimate all the models shown in this paper.

Statement of contribution

Larch, a freeware software package that can be used to estimate MNL, NL, and CNL model, significantly reduces estimation times for large, semi-aggregate datasets (in which the choice sets are not necessarily unique across decision makers). Benchmarking experiments against existing software and freeware packages showed that for complex discrete choice models, such as the ordered generalized extreme value model, estimation times were two seconds in Larch and four to five days in other packages.

Acknowledgement

The authors wish to thank Andrew Daly for providing access to ALOGIT for the purposes of testing the models in this paper, and for his invaluable help in translating the models and data for use with ALOGIT.

References

Abbe, E., Bierlaire, M., Toledo, T., 2007. Normalization and correlation of cross-nested logit models. Transp. Res. Part B Methodol. 41, 795–808.
ALOGIT 4.3 (2016), ALOGIT Software & Analysis Ltd.
Ben-Akiva, M.E., Lerman, S.R., 1985. Discrete Choice Analysis: Theory and Application to Travel Demand, vol. 9. MIT Press.
Berndt, E.R., Hall, B.H., Hall, R.E., Hausman, J.A., 1974. Estimation and inference in nonlinear structural models. In: Annals of Economic and Social Measurement, Volume 3, number 4. NBER, pp. 653–665.
Bierlaire, M., 2016. Pythonbiogeme: a Short Introduction. Report TRANSP-OR 160706. Transport and Mobility Laboratory, School of Architecture, Civil and Environmental Engineering, Ecole Polytechnique Fédérale de Lausanne, Switzerland.
Coldren, G., 2005. Modeling the Competitive Dynamic Among Air-travel Itineraries with Generalized Extreme Value Models. PhD thesis. Northwestern University.
Daganzo, C., 1979. Multinomial Probit: the Theory and its Application to Demand Forecasting. Academic Press.
Fosgerau, M., McFadden, D., Bierlaire, M., 2013. Choice probability generating functions. J. Choice Model. 8, 1–18.
Garrow, L.A., 2010. Discrete Choice Modelling and Air Travel Demand: Theory and Applications. Ashgate Publishing, Ltd.
Lurkin, V., Garrow, L.A., Higgins, M.J., Newman, J.P., Schyns, M., 2017. Accounting for price endogeneity in airline itinerary choice models: an application to continental us markets. Transp. Res. Part A Policy Pract. 100, 228–246.
McFadden, D., 1974. Conditional logit analysis of qualitative choice behavior. In: Frontiers in Econometrics, vols. 105&142. Academic Press, New York.
McFadden, D., 1978. Modeling the choice of residential location. Transp. Res. Rec. 673.
McFadden, D., 2001. Economic choices. Am. Econ. Rev. 91 (3), 351–378.
Small, K.A., 1987. A discrete choice model for ordered alternatives. Econ. J. Econ. Soc. 409–424.
Stata Statistical Software: Release 14. (2015), StataCorp LP, College Station, TX.
Train, K., 2003. 'Mixed Logit', Discrete Choice Methods with Simulation, pp. 139–154.
Vovsha, P., 1997. The cross-nested logit model: application to mode choice in the tel-aviv metropolitan area. Transp. Res. Rec. 1607, 6–15.
Wen, C.-H., Koppelman, F.S., 2001. The generalized nested logit model. Transp. Res. Part B Methodol. 35 (7), 627–641.
Williams, H.C., 1977. On the formation of travel demand models and economic evaluation measures of user benefit. Environ. Plan. A 9 (3), 285–344.

[9] Due to data non-disclosure agreements, we were unable to enlist Andrew Daly's assistance in preparing the data for these much larger models, and as a result this number may not be optimized to represent the best possible computational performance for ALOGIT.