CIRRELT

Centre interuniversitaire de recherche
sur les réseaux d'entreprise, la logistique et le transport

Interuniversity Research Centre
on Enterprise Networks, Logistics and Transportation

A Dynamic Programming Approach for Quickly Estimating Large Scale MEV Models

Tien Mai
Emma Frejinger
Mogens Fosgerau
Fabien Bastin

June 2015

CIRRELT-2015-24

Bureaux de Montréal :
Université de Montréal
Pavillon André-Aisenstadt
C.P. 6128, succursale Centre-ville
Montréal (Québec)
Canada H3C 3J7
Téléphone : 514 343-7575
Télécopie : 514 343-7121

Bureaux de Québec :
Université Laval
Pavillon Palasis-Prince
2325, de la Terrasse, bureau 2642
Québec (Québec)
Canada G1V 0A6
Téléphone : 418 656-2073
Télécopie : 418 656-2624

www.cirrelt.ca

A Dynamic Programming Approach for Quickly Estimating Large Scale MEV Models

Tien Mai[1], Emma Frejinger[1,*], Mogens Fosgerau[2], Fabian Bastin[1]

[1] Interuniversity Research Centre on Enterprise Networks, Logistics and Transportation (CIRRELT) and Department of Computer Science and Operations Research, Université de Montréal, P.O. Box 6128, Station Centre-Ville, Montréal, Canada H3C 3J7

[2] Technical University of Denmark, Denmark, and Royal Institue of Technology, Sweden

Abstract. In this paper we propose a way to estimate static Multivariate Extreme Value (MEV) models with very large choice sets in short computational time. Similar to the network MEV model (Daly and Bierlaire, 2006) the correlation structure is defined by a rooted, directed graph where each node without successor is an alternative. We show how to compute choice probabilities based on the graph using a dynamic programming approach. This allows us to estimate the models by maximum likelihood using the Nested Fixed Point algorithm proposed by Rust (1987). Moreover, we show that, under some conditions, the resulting models are consistent with MEV theory and generalize the network MEV model. We present numerical results based on simulated data with varying number of alternatives and nesting structures. We show that we can estimate large models, for example, a cross-nested model with 200 nests and 500,000 alternatives, 2,000,000 observations and 210 parameters needs between 100-200 iterations to converge (4.3 hours on an Intel(R) 3.2GHz machine using a non-parallelized code).

Keywords: Multivariate Extreme Value (MEV), dynamic programming, discrete choice, maximum likelihood estimation, nested fixed point, value iterations.

Acknowledgement: This research was partly funded by the National Sciences and Engineering Research Council of Canada (NSERC), discovery grant 435678-2013.

Results and views expressed in this publication are the sole responsibility of the authors and do not necessarily reflect those of CIRRELT.

Les résultats et opinions contenus dans cette publication ne reflètent pas nécessairement la position du CIRRELT et n'engagent pas sa responsabilité.

[*] Corresponding author: Emma.Frejinger@cirrelt.ca

Dépôt légal – Bibliothèque et Archives nationales du Québec
Bibliothèque et Archives Canada, 2015

1 Introduction

Dynamic discrete choice models are in general more complex to estimate and to apply than static discrete choice models. The reason is that dynamic programming problems need to be solved in order to evaluate the log-likelihood function. Recently, Fosgerau et al. (2013) and Mai et al. (2015) showed that a dynamic discrete choice formulation of the path choice problem is actually simpler to deal with than the classic, path based, static discrete choice model. This paper builds on a similar idea but in a different context. We propose a dynamic discrete choice approach that allows to estimate large Multivariate Extreme Value (MEV) models (McFadden, 1978) in short computational time.

The correlation structure of the alternative specific utilities is defined by a rooted, directed and connected graph where each node without successors is an alternative. Choice probabilities are defined by paths in this graph. In turn, path probabilities are computed by a dynamic discrete choice model, in a way similar to the route choice model proposed by Mai et al. (2015). This work is however different from a route choice setting since the graph corresponds to a correlation structure (not a transport network), has many destinations and more importantly, cost-less arcs (except for nesting parameters). Given some assumptions, the resulting choice model is equivalent to the network MEV model proposed by Daly and Bierlaire (2006). Our main objective is to estimate these models in short computational time. The main challenge lies in the definition and the computation of the expected maximum utility (value function) from a node in the graph to the nodes representing the alternatives and the computation of choosing alternatives as well as their gradients.

We use the nested fixed point algorithm proposed by Rust (1987) to estimate the model. The value functions are computed by using a value iteration method as in Mai et al. (2015). The choice probability of a given alternative is decomposed into sequences of node probabilities in the graph and we show that they can be computed by using the expected flows from the root to destinations, leading to a system of linear equations. In order to have efficient optimization algorithms for the maximum likelihood estimation we derive the derivatives of the value functions as well as the choice probabilities. We show that they are also solutions to linear systems of equations. Moreover, we derive demand elasticities. We present computational times for the estimation of a cross-nested and a network MEV models with different simulated data.

We make three main contributions. First we apply a dynamic discrete choice model to graphs of correlation structures and show that the model generalizes the network MEV model. Second, we propose efficient methods for the estimation of the model, i.e. a value iteration method to compute the value functions, systems of linear equations for the computations of the choice probabilities, gradients and elasticities. Third, the estimation code is implemented in MATLAB and is freely available upon request.

The paper is structured as follows. Section 2 presents a dynamic discrete choice approach for static discrete choice models. Section 3 provides an illustrative example using a cross-nested logit model. Section 4 discusses the properties of the model related to the MEV models and Section 6 derives the demand elasticities. We present the numerical results in Section 7 and finally Section 8 concludes.

2 A dynamic discrete choice approach for static discrete choice models

We consider a directed connected graph $\mathcal{G} = (\mathcal{N}, \mathcal{A})$ where $\mathcal{N}$ and $\mathcal{A}$ are the set of nodes and arcs, respectively. A subset of the nodes have no successors and define an alternative j in a choice set C. We assume that the graph does not have multiple links for any given node pair and there is only one node with no predecessor that we call the root. Considering the root as an origin, the nodes representing the alternatives can therefore be viewed as destinations in $\mathcal{G}$ and there are paths connecting the root and the destinations. The graph is used to define the correlation structure and a simple example is the well-known tree of a nested logit model where each leaf is an alternative.

For each node $k \in \mathcal{N}$, we denote the set of node successors $\mathcal{N}(k)$. The utility of node $a \in \mathcal{N}(k)$ conditional on its predecessor k is

$$u(a|k;\beta) = v(a|k;\beta) + \mu_k(\epsilon(a) - \gamma), \tag{1}$$

where $v(a|k;\beta)$ is a deterministic utility associated with a given k, β is a vector of parameters to be estimated, μ_k is a strictly positive scale parameter, $\epsilon(a)$ is extreme value type I and i.i.d over $a \in \mathcal{N}(k)$ and γ is Euler's constant. The Euler's constant is used in order to ensure that the random terms have zero mean. We note that the utilities can include nesting parameters and attributes of alternatives. This is different from the utilities considered in route choice applications (Fosgerau et al., 2013, Mai et al., 2015) where the graph is a road network, utilities are defined for arcs based on road attributes and there is only one path per observation.

The probability of $j \in C$ is the sum of the probabilities of all paths connecting r and j, and we denote the set of all such paths $\Omega(j)$. A path is defined by a sequence of nodes $k_0, k_1, \ldots, k_J$ such that $k_{i+1} \in \mathcal{N}(k_i)$, $\forall i = 0, \ldots, J-1$, where k_0 is the root r and k_J represents alternative j. Path probabilities are defined based on the probability of each node in the path and the probability of choosing

j over choice set C is

$$P(j) = \sum_{[k_0,\dots,k_J]\in\Omega(j)} \prod_{i=0}^{J-1} P(k_{i+1}|k_i), \quad j \in C \tag{2}$$

where $P(k_{i+1}|k_i)$ is the probability of node k_{i+1} given node k_i.

The key here is how to compute the node probabilities since they depend on nodes that are available downstream. Similar to Mai et al. (2015) this is the expected maximum utility (or value functions) $V(k)$ from a node k to the destinations. We assume each node $j \in C$ associates with an deterministic utility of the respective alternative U_j and we define $V(j) = U_j$. The model can be considered as an infinite horizon dynamic programming problem with absorbing states $j \in C$, thus the value function $V(k)$ for $k \in \mathcal{N}\backslash C$ is recursively defined by Bellman's equation

$$V(k;\beta) = \mathbb{E}\left[\max_{a\in\mathcal{N}(k)} \{v(a|k;\beta) + V(a;\beta) + \mu_k(\epsilon(a) - \gamma)\}\right], \forall k \in \mathcal{N}\backslash C \tag{3}$$

or equivalently

$$\frac{1}{\mu_k}V(k;\beta) = \mathbb{E}\left[\max_{a\in\mathcal{N}(k)} \left\{\frac{1}{\mu_k}\Big(v(a|k;\beta) + V(a;\beta)\Big) + \epsilon(a) - \gamma\right\}\right], \forall k \in \mathcal{N}\backslash C \tag{4}$$

which in this case is the logsum

$$\frac{1}{\mu_k}V(k;\beta) = \ln\Big(\sum_{a\in\mathcal{N}(k)} e^{\frac{1}{\mu_k}(v(a|k;\beta)+V(a;\beta))}\Big) \quad \forall k \in \mathcal{N}\backslash C \tag{5}$$

and for notational simplicity we also omit from now on β from the value functions V and the node-based utilities v. The probability of node a given node k is given by the MNL model

$$P(k|a) = \delta(a|k)e^{\frac{1}{\mu_k}(V(a)+v(a|k)-V(k))}, \quad \forall k, a \in \mathcal{N} \tag{6}$$

where $\delta(a|k)$ equals one if $a \in \mathcal{N}(k)$ and zero otherwise so that the probability is defined for all $a, k \in \mathcal{N}$. The utilities of other nodes in the network and the scale parameters μ_k, $k \in \mathcal{N}$, are related to the correlation structure.

According to (5), if we define a vector Y of size $|\mathcal{N}|$ with entries

$$Y_k = e^{\frac{V(k)}{\mu_k}}, \quad \forall k \in \mathcal{N} \tag{7}$$

then the value functions are the solutions to the following non-linear system

$$Y_k = \begin{cases} \sum_{a\in\mathcal{N}(k)} e^{v(a|k)/\mu_k} Y_a^{\mu_a/\mu_k} & \text{if } k \in \mathcal{N}\backslash C, \\ e^{U_k} & \text{if } k \in C. \end{cases} \tag{8}$$

Using (6), the choice probability of a node a given k can be written as

$$P(a|k) = \delta(a|k) \sum_{a\in\mathcal{N}(k)} e^{v(a|k)/\mu_k} \frac{Y_a^{\mu_a/\mu_k}}{Y_k}, \quad \forall k, a \in \mathcal{N} \tag{9}$$

It is convenient to compute choice probabilities using (2) and (6) because we only need to compute a vector of value functions of size $|\mathcal{N}|$ for each alternative. This can easily be done using the same techniques as for the route choice applications (Mai et al., 2015).

3 Illutrative example

In this section we use a cross-nested model as illustration. In this case the graph consists of three layers: a root r, a set of nodes M representing the nests and a set of nodes C representing the alternatives (see Figure 2).

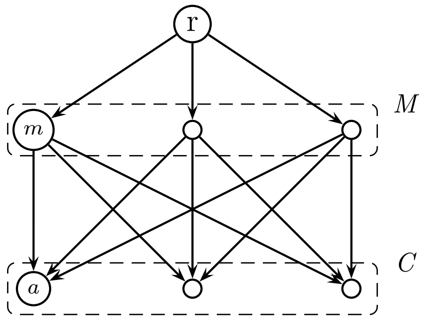

Figure 1: A cross-nested structure

There is an arc between the root and each nest and arcs between the nests and the alternatives. Assuming that α_{am} is the cross-nested parameters for a given nest m and an alternative a, and considering the network, given two nodes $m, a \in \mathcal{N}$, $m \in M$ and $a \in C$, we define $v(a|m) = \mu_m \ln(\alpha_{am})$ if $a \in \mathcal{N}(m)$. Moreover we assume that $v(m|r) = 0$, $\forall m \in M$. According to (5) and (6), given a nest $m \in M$

and alternative a, we have

$$P(a|m) = \frac{e^{(U_a+v(a|m))/\mu_m}}{\sum_{a'\in\mathcal{N}(m)} e^{(U_{a'}+v(a'|m))/\mu_m}}$$
$$= \frac{\alpha_{am}e^{U_a/\mu_m}}{\sum_{a\in C}\alpha_{am}e^{U_a/\mu_m}}$$

and

$$P(m|r) = \frac{e^{V(m)/\mu_r}}{\sum_{m'\in M} e^{V(m')/\mu_r}}.$$

Moreover, for each nest $m \in M$ the respective value function is given by (5) as

$$V(m) = \mu_m \ln\Big(\sum_{a\in\mathcal{N}(m)} e^{\frac{1}{\mu_m}(v(a|m)+V(a))}\Big). \tag{10}$$

Using (5), the choice probability is

$$P(a) = \sum_{m\in M} P(m|r)P(a|m)$$
$$= \sum_{m\in M} \frac{\big(\sum_{a\in C}\alpha_{am}e^{U_a/\mu_m}\big)^{\mu_m/\mu_r}}{\sum_{m\in M}\big(\sum_{a\in C}\alpha_{am}e^{U_a/\mu_m}\big)^{\mu_m/\mu_r}} \frac{\alpha_{am}e^{U_a/\mu_m}}{\sum_{a\in C}\alpha_{am}e^{U_a/\mu_m}}$$

which is equivalent to the choice probability given by Ben-Akiva and Bierlaire (1999). We note that other specifications of the cross-nest logit model can be obtained by defining a graph such that the choice probabilities are equivalent, for instance the paired combinatorial logit model (Koppelman and Wen, 2000), the generalised nested logit model Wen and Koppelman (2001), the ordered GEV model Small (1987), the link-nested logit model (Vovsha and Bekhor, 1998), the GenL model (Swait, 2001). In the next section we show that that if the graph $\mathcal{G}$ and scale parameters μ_k $\forall k \in \mathcal{N}$ satisfy some conditions, the resulting model is an additive random utility MEV model.

4 MEV consistency

We explore the properties of the model presented above by showing that under some conditions the resulting model is an additive random utility MEV model. We prove the MEV consistency for the case when the graph is cycle-free. (It is possible to extend this result graphs with cycles, it is work in progress.)

Theorem 1 *If the graph $\mathcal{G}$ is a non-empty, cycle-free and $\mu_k \geq \mu_a$, $\forall k, a \in \mathcal{N}, a \in \mathcal{N}(k)$, then the model is an additive random utility MEV model with the generating function $G(e^{U_i}, i \in C) = Y_r$, and Y_r is a $\frac{1}{\mu_r} - MEV$ function.*

Proof. The proof is based on the network MEV model (Daly and Bierlaire, 2006). We assume each arc (k, a), $a \in \mathcal{N}(k)$, is associated with a parameter $\alpha_{ka} = e^{v(a|k)/\mu_k}$ and each node $k \in \mathcal{N}$ associates with a positive scale parameter $\delta_k = 1/\mu_k$. We denote $y_k = e^{U_k}$, $\forall k \in C$. We also define

$$G^k(y_k) = y_k^{\delta_k}, \quad k \in C \tag{11}$$

and

$$G^k(y) = \sum_{a \in \mathcal{N}(k)} \alpha_{ka} G^a(y)^{\delta_k/\delta_a}, \forall k \in \mathcal{N}\backslash C \tag{12}$$

Daly and Bierlaire (2006) show that if $\delta_k \leq \delta_a$, $\forall a, k \in \mathcal{N}, a \in \mathcal{N}(k)$, the function $G^k()$ associated with a node $k \in \mathcal{N}$ is a $\delta_k - MEV$ function. This result allows the network MEV model to be consistent with McFadden's MEV theory and hence with additive random utility maximization (ARUM) (see for instance Mogens Fosgerau, 2013).

Therefore, we can prove the theorem by showing that

$$Y_k = G^k, \quad \forall k \in \mathcal{N}. \tag{13}$$

Indeed, according to (11)

$$Y_k = G^k, \quad \forall k \in C. \tag{14}$$

For $k \in \mathcal{N}\backslash C$, from (5) we have

$$V(k) = \mu_k \ln \left(\sum_{a \in \mathcal{N}(k)} e^{\frac{1}{\mu_k}(v(a|k)+V(a))} \right) \tag{15}$$

so, from (7),

$$Y_k = \sum_{a \in \mathcal{N}(k)} \alpha_{ka} e^{\frac{V(a)}{\mu_k}} = \sum_{a \in \mathcal{N}(k)} \alpha_{ka} Y_a^{\mu_a/\mu_k}$$

or equivalently

$$Y_k = \sum_{a \in \mathcal{N}(k)} \alpha_{ka} Y_a^{\delta_k/\delta_a}. \tag{16}$$

The result then follows from (11), (12), (14) and (16). In other words, Y_k associated with a node $k \in \mathcal{N}$ is a $\frac{1}{\mu_k} - MEV$ function.

We now analyze the probabilities given by a MEV model with the generating function $G(y) = G(y_i, i \in C) = Y_r$, where r is the root of the network. Each alternative i is associated with the utility $U_i + \epsilon_i$, where vector ϵ is MEV distributed with the generating function $G(y)$. Since Y_r is a $\frac{1}{\mu_r} - MEV$ function, the choice probability (McFadden, 1978) is

$$\begin{aligned} P(i) &= \frac{y_i \frac{\partial G}{\partial y_i}(y)}{1/\mu_r G(y)} \\ &= \frac{\mu_r y_i}{Y_r}\frac{\partial Y_r}{\partial y_i}, \quad i \in C \end{aligned} \tag{17}$$

From (8), the partial derivative of Y_k with respect to y_i, $i \in C$ is

$$\frac{\partial Y_k}{\partial y_i} = \sum_{a\in\mathcal{N}(k)} e^{v_{ka}/\mu_k}\frac{\mu_a}{\mu_k} Y_a^{\mu_a/\mu_k}\frac{\partial Y_a}{Y_a \partial y_i}, \quad k \in \mathcal{N}\backslash C. \tag{18}$$

Denote $S^i_k = \frac{\mu_k y_i}{Y_k}\frac{\partial Y_k}{\partial y_i}$, $i \in C, k \in \mathcal{N}$. Based on (42), we obtain a recursive formulas for S^i_k as

$$S^i_k = \sum_{a\in\mathcal{N}(k)} e^{v_{ka}/\mu_k}\frac{Y_a^{\mu_a/\mu_k}}{Y_k} S^i_a = \sum_{a\in\mathcal{N}(k)} P(a|k)S^i_a, \quad \forall k \in \mathcal{N}\backslash C. \tag{19}$$

Note that

$$S^i_i = \frac{\mu_i y_i}{Y_i}\frac{\partial Y_i}{\partial y_i} = \frac{\mu_i y_i}{y_i^{1/\mu_i}}\frac{1}{\mu_i} y_i^{1/\mu_i - 1} = 1,$$

so the choice probability given by the MEV model is

$$P(i) = S^i_r = \sum_{[k_l,\dots,k_0=i]\in\Omega(i)} \prod_{i=0}^{l-1} P(k_i|k_{i+1}), \quad \forall i \in C \tag{20}$$

which is equivalent to the probability given in (2). So basically when the graph is cycle-free the resulting model is equivalent to the network MEV model, so the properties presented in Daly and Bierlaire (2006) apply i.e. the model generalizes many MEV models proposed in the literature. ■

In the following section we present how to estimate the model by using the nested fixed point algorithm (Rust, 1987).

5 Maximum likelihood estimation

There are different ways to estimate a dynamic discrete choice model (see for instance Aguirregabiria and Mira, 2010). We use the nested fixed point algorithm proposed by Rust (1987). The algorithm combines an outer iterative non-linear optimization algorithm searching over the parameter space with an inner algorithm solving the value functions. The computation of the choice probabilities as well as the likelihood requires the value functions. In the following we discuss how to solve the value functions, compute the choice probabilities and the log-likelihood function.

5.1 Solving the the value functions

The probability of a path in the graph can be computed by using the value functions based on (2) and (6). In the following we describe a simple value iteration method which is efficient for our problem.

We define a matrix M of size $|\mathcal{N}| \times |\mathcal{N}|$ and a vector b of size $|\mathcal{N}|$, with entries

$$M_{ka} = \delta(a|k)e^{v(a|k)/\mu_k}; \quad b_k = \begin{cases} e^{U_k} & \text{if } k \in C \\ 0 & \text{if } k \in \mathcal{N}\backslash C \end{cases} \tag{21}$$

and a matrix $X(Y)$ of size $|\mathcal{N}| \times |\mathcal{N}|$, with entries $X(Y)_{ka} = Y_a^{\mu_a/\mu_k}$, $\forall k, a \in \mathcal{N}$. The non-linear system (8) can be written as

$$Y_k = \sum_{a \in \mathcal{N}} M_{ka} Y_a^{\mu_a/\mu_k} + b_k, \quad \forall k \in \mathcal{N} \tag{22}$$

or equivalently

$$Y = [M \circ X(Y)]e + b \tag{23}$$

where $\circ$ is the element-by-element operation and e is a vector of size $|\mathcal{N}|$ with value one for all nodes. This equation can be solved by a value iteration. We start with an initial vector Y^0 and compute a new vector for each iteration i

$$Y^{i+1} \leftarrow [M \circ X(Y^i)]e + b. \tag{24}$$

In general, we iterate until a fixed point is found using $||Y^{i+1} - Y^i||^2 < \tau$ for a given threshold $\tau > 0$ as stopping criteria. It can be shown that if the Bellman's equation has a solution, the value iteration method converges after a finite number of iterations (see for instance Rust, 1987, 1988). Mai et al. (2015) use value iteration with dynamic accuracy to efficiently compute the vector of the value functions in a real road network which contains cycles. In a cycle-free graph it can be shown that the value iteration method converges to the fixed point solution after few iterations i.e. there exist $K > 0$ such that $Y^{i+1} = Y^i$, $\forall i > K$. For instance, in the case of cross-nested logit models, the value iterations only needs 3 iterations to converge, independently of the number of nodes and the structure of the graph.

5.2 Choice probabilities

Based on (2), the choice probability of a given alternative can be computed by enumerating all the paths connecting the root r with the destination $i \in C$ representing the alternative. This can be cumbersome if the graph is dense, or contains cycles, or the number of observations is large. In order to compute the choice probabilities in short computational time we use a method from route choice applications. Namely, we compute the flows in the graph from the root (origin) to destinations.

We consider the graph as a road network. We denote the demand for trips originating at node $k \in \mathcal{N}$ as $D(k)$. Denote the expected flow on node a as $F(a)$. This comprises the flow that originates on a and the expected incoming flow, so we have

$$F(a) = D(a) + \sum_{a \in \mathcal{N}} P(a|k)F(k)$$

and equivalently in matrix form as

$$(I - P^T)F = D \tag{25}$$

where I is the identity matrix, P is a matrix of size $|\mathcal{N}| \times |\mathcal{N}|$ with elements $P_{ka} = P(a|k)$, $\forall k, a \in \mathcal{N}$. This leads to

$$F = (I - P^T)^{-1}D. \tag{26}$$

In order to obtain the choice probabilities we define an origin specific demand vector D with zero-valued elements except for the root which equals one. The expected flows can be written as

$$F(k) = \begin{cases} \sum_{\substack{\{h_0,\dots,h_l\} \\ h_0=r, h_l=k \\ h_{t+1} \in \mathcal{N}(h_t), t=0,\dots,l-1}} \prod_{t=0}^{l-1} P(h_{t+1}|h_t) & \text{if } k \in \mathcal{N}\backslash\{r\} \\ 1 & \text{if } k = r \end{cases} \tag{27}$$

and note that according to (2) the probability of choosing an alternative is

$$P(i) = F(i), \forall i \in C, \tag{28}$$

where $i \in C$ is the destination representing the alternative. So the choice probabilities for all alternatives are the solutions to the the system of linear equations in (26).

5.3 Estimation

We assume that the utilities associated with alternatives, the deterministic utility associated with a pair of nodes $v(a|k)$, $a \in \mathcal{N}(k)$ and the scales of the model μ_k, $k \in \mathcal{N}$ are functions of parameters β to be estimated. The log-likelihood function, defined over the set of observations $n = 1, \ldots, N$, is

$$LL(\beta) = \sum_{n=1}^{N} P(i_n|C_n) \tag{29}$$

where i_n is the chosen alternative and C_n the the choice set with respect to individual n. The choice probability is defined in (2) and can be computed efficiently using (26) and (28).

For the maximum likelihood estimation, the network $\mathcal{G}$ generates MEV models with many parameters. That is, the scale parameters associated with each node μ_k, $k \in \mathcal{N}$ and the utility $v(a|k)$ for each node pair (k, a) in the network. Not all of them are identifiable from data. The scale parameters μ_k relevant only in terms of their ratio, exactly as for the nested logit model. Inspired by the cross-nested logit model, conditions for the utilities $v(a|k)$ would be (see for instance Daly and Bierlaire, 2006, Papola, 2004)

$$\kappa(a) = \sum_{k \in W(a)} e^{v(a|k)/\mu_k} = 1, \quad \forall a \in \mathcal{N}\backslash\{r\} \tag{30}$$

where $W(a)$ is the set of all predecessor nodes of a, $\forall a \in \mathcal{N}\backslash\{r\}$. Furthermore, according to Theorems 1 and **??** the constraints $\mu_k \geq \mu_a > 0$, $\forall a \in A(k)$ need to be satisfied for the MEV consistency. In summary, the maximum log-likelihood estimation can be formulated as a constrained non-linear optimization problem as

$$\max_{\substack{\mu_k \geq \mu_a > 0, \forall a \in A(k) \\ \kappa(a)=1, \forall a \in \mathcal{N}\backslash\{r\}}} LL(\beta). \tag{31}$$

Efficient nonlinear techniques for the problem require analytical derivatives of the log-likelihood function. They are provided in Appendix A. We note that the derivatives of the log-likelihood function can be computed efficiently by solving systems of linear equations.

6 Demand responses and elasticities

The elasticity of demand for alternative i with respect to an attribute x_j of alternative j is

$$e_{i,x_j} = \frac{\partial P(i)}{\partial x_j} \frac{x_j}{P(i)} = \frac{\partial P(i)}{\partial U_j} \frac{\partial U_j}{\partial x_j} \frac{x_j}{P(i)}, i, j \in C. \tag{32}$$

If the utility U_j is linear in x, $\frac{\partial U_j}{\partial x_j}$ is a constant. We now analyze the model structures in terms of the responses of demand to changes in the utility of alternatives $\frac{\partial P(i)}{\partial U_j}$. Similar to the previous section we derive formulas for the elasticity of demand so that they can be computed efficiently.

Note that $\frac{\partial P(i)}{\partial U_j} = \frac{\partial F_i}{\partial U_j}$ and the Jacobian of vector F with respect to U_j can be derived using (26)

$$\frac{\partial F}{\partial U_j} = (I - Q)^{-1} \frac{\partial Q}{\partial U_j} F, \tag{33}$$

where $Q = P^T$. Using (38), the derivative of an element Q_{ak}, $k, a \in \mathcal{N}$ with respect to U_j is

$$\frac{\partial Q_{ak}}{\partial U_j} = Q_{ak} \Big(\frac{\phi_{ka}}{Y_a} \frac{\partial Y_a}{\partial U_j} - \frac{\partial Y_k}{Y_k \partial U_j} \Big) \tag{34}$$

and hence requires the first derivatives of Y_k, $\forall k \in \mathcal{N}$, with respect to U_j. Taking the derivative of (22) with respect to U_j we obtain

$$\frac{\partial Y_k}{\partial U_j} = \sum_{a \in \mathcal{N}} \phi_{ka} M_{ka} Y_a^{\phi_{ka}-1} \frac{\partial Y_a}{\partial U_j} + \frac{\partial b_k}{\partial U_j}, \quad \forall k \in \mathcal{N}. \tag{35}$$

And we note that

$$\frac{\partial b_k}{\partial U_j} = \begin{cases} 0 & \text{if } k \neq j \\ Y_j/\mu_j & \text{if } k = j \end{cases}.$$

So if we denote a matrix $T(|\mathcal{N}| \times |\mathcal{N}|)$ with entries $T_{ka} = \phi_{ka} M_{ka} Y_a^{\phi_{ka}-1}$, $\forall k, a \in \mathcal{N}$, then the Jacobian of vector Y can be written as system of linear equations

$$\frac{\partial Y}{\partial U_j} = (I - T)^{-1} d \tag{36}$$

where d is a vector of size $|\mathcal{N}|$ with zero values for all nodes except for node j that equals Y_j/μ_j. Therefore, the elasticity of demand for alternative i with respect to an attribute x_j can be computed by solving the linear systems (33) and (36).

7 Numerical results

In this section we report the performance of the new approach based on simulated data with the purpose to evaluate computational times when estimating MEV models with very large choice sets. We first provide the performance results for a cross-nested logit model and then a network MEV model based on simulated data sets. Our code is implemented in MATLAB 2015 and we have used an Intel(R) machine, CPU 3.20GHz, running Window 8, 64-bit Operating system, x64-based processor. The machine has a multi-core processor but we only use one processor to estimate the model as the code has not been parallelized.

7.1 Cross-nested logit models

We consider the cross nested logit model presented in Section 3. The network contains a set of nests which connect to the alternatives and a root that connects to all the nest as in Figure 2.

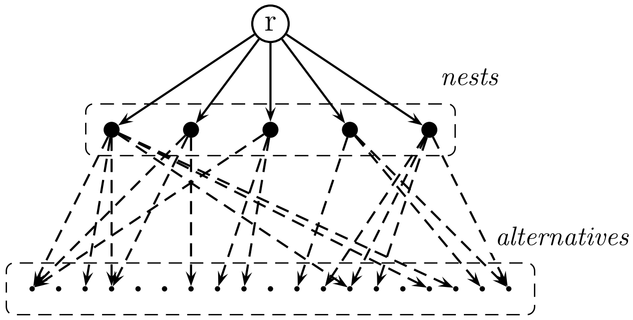

Figure 2: A cross-nested structure

We generate choice sets of sizes 10000, 100000 and 500000. The attributes are generated uniformly in interval $[0, 5]$ and we note that the alternative utilities are individual independent. We use a cross-nested logit model with 5 and 200 nests. In this application we estimate 6 parameters associated with alternative utilities and the scale parameters μ_k, $k \in \mathcal{N}\backslash C$. We estimate 4 α parameters associated with two different alternatives, the other parameters are fixed to their true values. For the optimization we use the interior point algorithm with BFGS to solve the constrained optimization problems. There are some models with large number of parameters to be estimated i.e. more than 200, we use the limited memory BFGS algorithm (L-BFGS) (for instance Nocedal and Wright, 1999, Chapter 9) to solve the large-scale problems. For the data with 5 and 200 nests, the optimization algorithms need around 100 to 300 iterations to converge. We report the data sets, and the computational time to compute the LL function and its gradient, for computing the elasticities for a given alternative and the total estimation time in Tables 1 and 2. For the data with 10000 alternatives we need around 5 seconds to estimate the model with 5 nests and less than 2 minutes for estimating the model with 200 nests. For the largest data (500000 alternatives), the estimation times are about 20 minutes for estimating the model with 5 nests and around 4 hours for the model with 200 nests. We note that the computational times for solving the elasticities for a given alternative are small (few seconds for the most complicated case).

Data	Model	# alters	# nests	# arcs in network	# Obs
D1	M1	10^4	5	17262	10^5
D1	M2	10^4	200	30011	10^5
D2	M3	10^5	5	172376	10^6
D2	M4	10^5	200	298237	10^6
D3	M5	5×10^5	5	862597	2×10^6
D3	M6	5×10^5	200	1490207	2×10^6

Table 1: Simulated data sets

			Computational time		
Data	Model	# params	LL and gradient	Estimation	Elasticities
D1	M1	16	0.14	4.61	0.04
D1	M2	210	1.44	84.62	0.06
D2	M3	16	1.75	301.55	0.48
D2	M4	210	18.29	1073.93	0.59
D3	M5	16	7.84	1462.9	2.27
D3	M6	210	88.52	15566.32	3.08

Table 2: Computational time (in seconds)

7.2 Multi-level cross-nested logit models

In this section we provide numerical results for a multi-level cross nested logit (or network MEV) model. Figure 3 shows the correlation structure given by the model, where there is a root connecting with 1^{st}-level nests and the 1^{st}-level nests connect with the nests in 2^{nd} level.

We use the data sets generated in the previous section. The model is defined with 5 nests in the first level and 50 nests in the second level. The nests in the first level connect to all the nests in the second level. We estimate 56 scale parameters μ_k, $k \in \mathcal{N}\backslash C$ and all parameters α_{ka}, $a \in \mathcal{N}(k)$ and $k, a \in \mathcal{N}\backslash C$. In total the model has 305 constraints and 312 parameters to be estimated. The networks representing the multi-level cross-nested models are more dense compared to the cross-nested model and the optimization algorithms require from 300 to 500 iterations to converge. Note that if we do not estimate the scale parameters μ_k, the optimization algorithm needs less than 50 iterations to converge. We report the computational times in Table 3. The results show that we can estimate the multi-level cross-nested logit models with large choice sets in reasonable times (about 14 hours for the model with 200 nests and the data with 500000 alternatives). The computational times for the multi-level cross-nested logit models are twice the

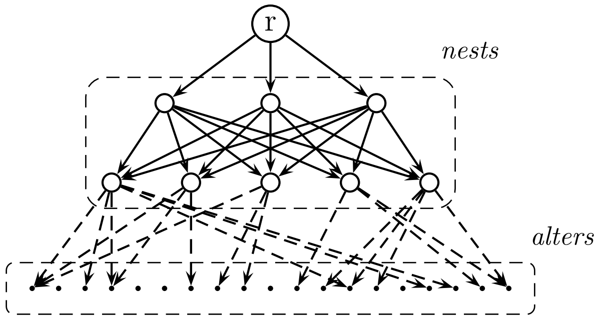

Figure 3: A tree logit structure

cross-nested logit models considered in the previous section. It is important to note that in the simulated data sets, the utilities are not individual specific. The number of observations therefore does not affect much the computational times. For some real problems, for instance mode-destination choice models, the alternative utilities depend on each individual and the value functions are individual specific, leading to a more expensive log-likelihood function. The estimation is therefore more costly, especially when dealing with large numbers of observations.

Data	# arcs	Computational time		
		LL and gradient	Estimation	Elasticities
D1	56972	2.74	715.6	0.12
D2	566751	38.47	11464.06	1.14
D3	2831668	162.52	52168.92	5.52

Table 3: Computational time (in seconds)

8 Conclusion

In this paper we have introduced a novel approach for the estimation of static discrete choice models based on the graph of correlation structure and the dynamic discrete choice framework. We have shown that under some conditions the resulting models is consistent with McFadden's MEV theory and equivalent to the network MEV model. We show how large scale MEV models can be estimated by maximum likelihood using the nested fixed point algorithm. Choice probabilities can be easily computed using the expected flows in the graph.

We have presented numerical experiments using simulated data. The results indicate that we are able to very quickly estimate the cross-nested and multi-levels cross-nested models with large choice sets and large number of observations. The estimation code is implemented in MATLAB and is available upon request.

Acknowledgement

This research was partly funded by the National Sciences and Engineering Research Council of Canada, discovery grant 435678-2013.

A Derivatives of the log-likelihood function

In this appendix, we derive the derivatives of the log-likelihood function defined in (29). The gradient of the choice probability $P(i_n|C_n)$ can be obtained by taking the Jacobian of vector F which can be derived based on (26). The Jacobian of F with respect to parameter β_j is

$$\frac{\partial F}{\partial \beta_j} = (I - Q)^{-1} \frac{\partial Q}{\partial \beta_j} F \tag{37}$$

where we denote $Q = P^T$ for notational simplicity. Hence it requires the first derivative of each element of matrix Q with respect to parameter β_j. Note that

$$Q_{ak} = P(a|k) = M_{ka} \frac{Y_a^{\mu_a/\mu_k}}{Y_k}, \quad \forall k, a \in \mathcal{N}. \tag{38}$$

We define $\phi_{ka} = \mu_a/\mu_k$ and take the derivative of a given Q_{ak} and obtain

$$\begin{aligned} \frac{\partial Q_{ak}}{\partial \beta_j} &= \frac{\partial M_{ka}}{\partial \beta_j} \frac{Y_a^{\phi_{ka}}}{Y_k} - M_{ka} \frac{Y_a^{\phi_{ka}}}{Y_k^2} \frac{\partial Y_k}{\partial \beta_j} \\ &+ M_{ka} \frac{Y_a^{\phi_{ka}}}{Y_k} \Big(\frac{\partial \phi_{ka}}{\partial \beta_j} \ln Y_a + \frac{\phi_{ka}}{Y_a} \frac{\partial Y_a}{\partial \beta_j} \Big). \end{aligned} \tag{39}$$

Hence it requires the derivative of Y_k, $\forall k \in \mathcal{N}$. We take the derivative of a given value Y_k, $k \in \mathcal{N}$ as defined by (22) and obtain

$$\frac{\partial Y_k}{\partial \beta_j} = \sum_{a \in \mathcal{N}} \left(\frac{\partial M_{ka}}{\partial \beta_j} Y_a^{\phi_{ka}} + M_{ka} Y_a^{\phi_{ka}} \Big(\frac{\partial \phi_{ka}}{\partial \beta_j} \ln Y_a + \frac{\phi_{ka}}{Y_a} \frac{\partial Y_a}{\partial \beta_j} \Big) \right) + \frac{\partial b_k}{\partial \beta_j}. \tag{40}$$

We introduce two matrices S and H of size $|A| \times |A|$ which has entries

$$\begin{cases} S_{ka} = \frac{\partial M_{ka}}{\partial \beta_j} Y_a^{\phi_{ka}} + M_{ka} Y_a^{\phi_{ka}} \frac{\partial \phi_{ka}}{\partial \beta_j} \ln Y_a \\ H_{ka} = M_{ka} Y_a^{\phi_{ka}} \frac{\phi_{ka}}{Y_a} \end{cases} \quad \forall k, a \in \mathcal{N}. \tag{41}$$

So (40) becomes

$$\frac{\partial Y_k}{\partial \beta_j} = \frac{\partial b_k}{\partial \beta_j} + \sum_{a\in\mathcal{N}(k)} \Big(S_{ka} + H_{ka}\frac{\partial Y_a}{\partial \beta_j}\Big), \quad \forall k \in \mathcal{N}. \tag{42}$$

This allows us to define the Jacobian of vector Y as a system of linear equation

$$\frac{\partial Y}{\partial \beta_j} = Se + H\frac{\partial Y}{\partial \beta_j} + \frac{\partial b}{\partial \beta_j} \Rightarrow \frac{\partial Y}{\partial \beta_j} = (I-H)^{-1}(Se + \frac{\partial b}{\partial \beta_j}) \tag{43}$$

which looks complicated but efficient to use to compute the gradient of $Y_k, \forall k \in \mathcal{N}$. Nevertheless, as suggested by Mai et al. (2015) we can derive the Jacobian of V instead of Y to avoid numerical issues. Note that $Y_k = e^{V(k)/\mu_k}$, the gradient of Y_k with respect to β_j can be written as

$$\frac{\partial Y_k}{\partial \beta_j} = \frac{\partial V(k)}{\partial \beta_j}\frac{Y_k}{\mu_k} - \frac{\partial \mu_k}{\partial \beta_j}\frac{V(k)Y_k}{\mu_k^2} \quad \forall k \in \mathcal{N}. \tag{44}$$

Using (18) we get

$$\frac{\partial V(k)}{\partial \beta_j} = \sum_{a\in\mathcal{N}} R_{ka} + \sum_{a\in\mathcal{N}} L_{ka}\frac{\partial V(a)}{\partial \beta_j} + h_k \quad \forall k \in \mathcal{N}, \tag{45}$$

where

$$\begin{aligned}
R_{ka} &= \mu_k \frac{\partial M_{ka}}{\partial \beta_j}\frac{Y_a^{\phi_{ka}}}{Y_k} + \mu_k M_{ka} Y_a^{\phi_{ka}} \frac{\partial \phi_{ka}}{Y_k \partial \beta_j}\ln Y_a - M_{ka}\frac{V(a)Y_a^{\phi_{ka}}}{\mu_k Y_k}\frac{\partial \mu_a}{\partial \beta_j}\\
L_{ka} &= M_{ka}\frac{Y_a^{\phi_{ka}}}{Y_k}\\
h_k &= \frac{\mu_k}{Y_k}\frac{\partial b_k}{\partial \beta_j} + \frac{V(k)}{\mu_k}\frac{\partial \mu_k}{\partial \beta_j}.
\end{aligned}$$

We denote R,L and h be three matrices and vector of size $|\mathcal{N}|\times|\mathcal{N}|$, $|\mathcal{N}|\times|\mathcal{N}|$, $|\mathcal{N}|$, with entries R_{ka}, L_{ka} and h_k, $\forall k, a \in \mathcal{N}$, respectively. The Jacobian of the vector of value functions can be written as a linear system

$$\frac{\partial V}{\partial \beta_j} = (I-L)^{-1}(Re + h). \tag{46}$$

Although (46) and (43) are theoretically equivalent, we now discuss the numerical differences between the two formulas. We consider the definition of matrix L where each element is defined as $L_{ka} = M_{ka}\frac{Y_a^{\phi_{ka}}}{Y_k}$. According to (22) we have $Y_k > M_{ka}Y_a^{\phi_{ka}} > 0$, leading to the fact that the elements of L vary in (0,1).

However, each element of H is $H_{ka} = M_{ka}Y_a^{\phi_{ka}-1}\phi_{ka}$ which varies in $(0,\infty)$. So the the elements of matrix L are closer in value, compared to matrix H, meaning that using (46) to compute the gradient of LL function is better than (43) for numerical reasons. Note that Mai et al. (2015) has a similar conclusion when comparing two formulas of the derivative of the value functions in route choice applications.

We note that the derivative of each element of matrix M with respect to parameter β_j is

$$\frac{\partial M_{ka}}{\partial \beta_j} = \delta(a|k)e^{\frac{v(a|k)}{\mu_k}}\left(\frac{\partial v(a|k)}{\mu_k \partial \beta_j} - v(a|k)\frac{\partial \mu_k}{\mu_k^2 \partial \beta_j}\right), \quad \forall k, a \in \mathcal{N}.$$

In summary, the derivatives of the model have complicated form but can be computed efficiently for large-scale problems using the linear systems in (37) and (46). The model derivatives allow us to use classic Hessian approximations such as BHHH and BFGS (see for instance Berndt et al., 1974, Nocedal and Wright, 2006) to efficiently maximize the log-likelihood function.

References

Aguirregabiria, V. and Mira, P. Dynamic discrete choice structural models: A survey. *Journal of Econometrics*, 156(1):38–67, 2010.

Ben-Akiva, M. and Bierlaire, M. Discrete choice methods and their applications to short-term travel decisions. In Hall, R., editor, *Handbook of Transportation Science*, pages 5–34. Kluwer, 1999.

Berndt, E. K., Hall, B. H., Hall, R. E., and Hausman, J. A. Estimation and inference in nonlinear structural models. *Annals of Economic and Social Measurement*, 3/4:653–665, 1974.

Daly, A. and Bierlaire, M. A general and operational representation of generalised extreme value models. *Transportation Research Part B: Methodological*, 40(4): 285 – 305, 2006.

Fosgerau, M., Frejinger, E., and Karlström, A. A link based network route choice model with unrestricted choice set. *Transportation Research Part B*, 56:70–80, 2013.

Mai, T., Fosgerau, M., and Frejinger, E. A nested recursive logit model for route choice analysis. *Transportation Research Part B: Methodological*, 75(0):100 – 112, 2015. ISSN 0191-2615.

McFadden, D. Modelling the choice of residential location. In Karlqvist, A., Lundqvist, L., Snickars, F., and Weibull, J., editors, *Spatial Interaction Theory and Residential Location*, pages 75–96. North-Holland, Amsterdam, 1978.

Mogens Fosgerau, A. K. Emma Frejinger. A link based network route choice model with unrestricted choice set. *Transportation Research Part B: Methodological*, 56(0):70 – 80, 2013. ISSN 0191-2615. doi: http://dx.doi.org/10.1016/j.trb.2013.07.012. URL `http://www.sciencedirect.com/science/article/pii/S0191261513001276`.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, NY, USA, 1999.

Nocedal, J. and Wright, S. J. *Numerical Optimization*. Springer, New York, NY, USA, 2nd edition, 2006.

Papola, A. Some developments on the cross-nested logit model. *Transportation Research Part B*, 38(9):833–851, 2004.

Rust, J. Optimal replacement of GMC bus engines: An empirical model of Harold Zurcher. *Econometrica*, 55(5):999–1033, 1987.

Rust, J. Maximum likelihood estimation of discrete control processes. *SIAM Journal on Control and Optimization*, 26(5):1006–1024, 1988.

Small, K. A. A discrete choice model for ordered alternatives. *Econometrica*, 55 (2):409–424, 1987.

Wen, C.-H. and Koppelman, F. The generalized nested logit model. *Transportation Research Part B*, (35):627–641, 2001.