Estimating Large-Scale Tree Logit Models

Srikanth Jagabathula
Stern School of Business, New York University, New York, NY 10012, sjagabat@stern.nyu.edu
Paat Rusmevichientong
Marshall School of Business, University of Southern California, Los Angeles, CA 90089, rusmevic@marshall.usc.edu
Ashwin Venkataraman
Naveen Jindal School of Management, University of Texas at Dallas, Richardson, TX 75080, axv190029@utdallas.edu
Xinyi Zhao
Stern School of Business, New York University, New York, NY 10012, xz2197@stern.nyu.edu

May 7, 2022

We describe an efficient estimation method for large-scale tree logit models, using a novel change-of-variables transformation that allows us to express the negative log-likelihood as a strictly convex function in the leaf node parameters and a difference of strictly convex functions in the non-leaf node parameters. Exploiting this representation, we design a fast iterative method that computes a sequence of parameter estimates using simple closed-form updates. Our algorithm relies only on first-order information (function and gradients values), but unlike other first-order methods, it does not require any step size tuning or costly projection steps. The sequence of parameter estimates yields increasing likelihood values, and we establish sublinear convergence to a stationary point of the maximum likelihood problem. Numerical results on both synthetic and real data show that our algorithm outperforms state-of-the-art optimization methods, especially for large-scale tree logit models with thousands of nodes.

Key words: tree logit, choice modeling, parameter estimation

1. Introduction

Introduced by McFadden (1981), the tree logit model—also referred to as the nested, d-level nested, sequential, hierarchical, structured, or multi-level logit model—is a versatile discrete choice model used to capture customer choice behavior in a range of practical applications. The model is best suited for applications in which products are naturally grouped by their characteristics into a product hierarchy or taxonomy. For example, Figure 1 shows a grouping of flight itineraries based on the departure time (morning or evening) and the number of stops (non-stop or 1-stop). When choosing from a product hierarchy, customers tend to view the products within the same sub-division as more similar than those in other sub-divisions; hence, customers view a morning flight as more similar to another morning flight than to an evening one. Consequently, if a morning flight is sold out, the customer is more likely to substitute another morning flight than to substitute an evening one. Such product similarities, arising by virtue of the hierarchy, are naturally captured

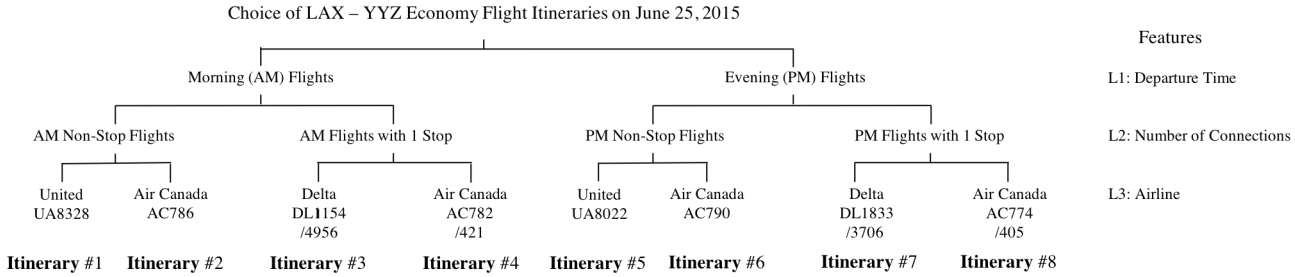


Figure 1 A collection of one-way flight itineraries from LAX to YYZ organized in a product hierarchy based on the departure time (morning or evening) and the number of stops (non-stop or 1-stop). The product hierarchy influences how customers choose, because products within the same sub-division (e.g., morning flights) are generally viewed as more similar than those in different sub-divisions (e.g., evening flights). The data were collected on March 1, 2015, from Orbitz.com.

by the tree logit model. This model represents the hierarchy as a rooted tree graph (hence the name). It assumes that in each choice instance, a customer samples product utilities according to some distribution and then chooses the highest-utility product from among the offered products. The tree logit model captures product similarities by allowing the utilities of products within the same sub-division of the hierarchy to be correlated. By contrast, the more popular, but simpler, multinomial logit (MNL) model cannot capture correlations among product utilities.[1]

Because of its versatility, the tree logit model has been adopted by researchers to represent customer choices in a diverse range of applications, including the choice of transportation modes (Hensher 1986, Forinash and Koppelman 1993), household locations (Anas 1982), automobiles (Train 1980, 1988), and phone-usage levels (Lee 1999); see Koppelman and Bhat (2006) for an overview. The model has also recently gained popularity within the operations community when it was observed that in addition to its versatility, its structure is sufficient to enable key operational decisions to be solved in a computationally tractable manner. Researchers have exploited this structure to investigate variants of the price and assortment optimization problems, such as price optimization with and without competition (Li and Huh 2011, Gallego and Wang 2014), under quality consistency constraints (Davis et al. 2016), and in the presence of bound constraints (Rayfield et al. 2015); joint product line design and pricing (Li 2018, Chapter 4); joint assortment and price optimization under a specific tree logit model for monopolistic and competitive settings (Kök and Xu 2011); assortment optimization under a two-level tree logit model[2] without constraints (Davis

[1] The MNL thus suffers from the well-documented "independence of irrelevant alternatives" limitation. It was to address this limitation that McFadden (1981) introduced the tree logit model as a generalization of the MNL.

[2] A two-level tree logit model is one in which the distance from the root node to any leaf node is at most two.

et al. 2014) and with cardinality and capacity constraints (Gallego and Topaloglu 2014, Feldman and Topaloglu 2015); assortment and price optimization under a tree logit model without constraints (Li et al. 2015); and online personalized resource allocation (Gallego et al. 2018).

Despite the popularity of tree logits across a range of disciplines, surprisingly, the model still lacks a scalable estimation method. Existing estimation methods do not scale to practical problem instances, which can involve hundreds, or even thousands, of products. As a result, even though we have access to decision methods that can be applied to thousands of products organized into complicated tree structures, they are rendered useless because we cannot estimate the model parameters.

Most published work has fitted the tree logit models to only tens of products. The main difficulty lies in maximizing the data log-likelihood function. The maximization problem is typically formulated as a constrained non-convex optimization problem, and the prevailing approach, in both academia and practice, has been to solve this problem using some general-purpose non-linear optimization method offered by one of the commercial solvers. These methods do not scale to large problem sizes because they are not designed to meaningfully exploit any special structure of the tree logit models. Even with small numbers of products, they suffer from numerical issues, such as getting stuck in bad local optima or obtaining solutions inconsistent with the random utility maximization principle. Indeed, Koppelman and Bhat (2006) noted that with generic optimization algorithms in commercial software packages, "very different starting points can eventually settle to the same solution while other, nearby starting points, can diverge to different results" and "some or all (or none) of [these results] may be consistent with utility maximization." Furthermore, none of the existing methods offers theoretical performance guarantees for tree logits.

Our key contribution is to develop an estimation method that scales to thousands of products. Tree logit models are described by two types of parameters: the mean utility parameters and the nest dissimilarity parameters. Through a variable transformation, we transform the constraints into simple non-negativity constraints and the non-convex log-likelihood function into a function that is convex in the mean utility parameters and a difference of strictly convex functions in the next dissimilarity parameters. We then exploit the resulting structure to design an efficient majorization-minimization (MM) (Hunter and Lange 2004) algorithm.[3] We establish strong theoretical guarantees for the resulting solutions, and empirically demonstrate the performance of our proposed method on both synthetic and real data. Next, we summarize our key contributions.

[3] A special case of the majorization-minimization principle is the expectation-maximization algorithm (Dempster et al. 1977), which has been used to estimate the parameters of various choice models; see, for example, Vulcano et al. (2012) and Şimşek and Topaloglu (2018).

1.1 Contributions

The main contributions of our paper are as follows.

Scalable MM algorithm. Through the use of MM techniques and a novel change of variables (see Section 4.1), we reduce the likelihood problem into solving a sequence of optimization problems, each with a simpler surrogate objective function. We show that each of these simpler optimization problems can be solved in closed form (see Theorems 4.3 and 4.8), resulting in a sequence of solutions whose associated log-likelihood value is guaranteed to be increasing. Our algorithm relies only on first-order information (function and gradients values), but unlike other first-order methods, it does not require any step size tuning or costly projection steps, allowing it to scale to large problem instances with thousands of products. We also show that the standard MM algorithm (Hunter 2004, Abdallah and Vulcano 2021) for estimating the parameters of the MNL model can be obtained as a special case of our algorithm; see the discussion after Theorem 4.8.

Convergence rate guarantees: We derive strong performance guarantees for our method. In particular, we show that the sequence of parameter estimates generated by our algorithm converges to a stationary point of the maximum likelihood problem at a *sub-linear rate* in the number of iterations; see Theorem 4.4 and Theorem G.1 in Appendix G. Compared to the standard gradient descent (GD) algorithm, our algorithm can be a factor n faster (Theorem 4.6), where n is the number of products. To the best of our knowledge, our result is the first to establish a stronger convergence rate guarantee for the MM algorithm over the standard GD algorithm. Prior to our result, the best known result in the literature was from Vojnovic et al. (2020), who established[4] that both MM and GD algorithms have the same sub-linear convergence rates (modulo constant factors), but only for the special case of the Bradley-Terry model (MNL with pairwise comparisons). Apart from this result, there isn't substantial work on analyzing the convergence rates of the MM algorithm. Most of the literature (Hunter 2004, Abdallah and Vulcano 2021), however, has observed that the MM algorithm converged faster than gradient descent and off-the-shelf optimization routines in practice for the MNL model. Our result now provides theoretical support for this empirical observation, especially in large-scale settings.

Excellent empirical performance on large-scale problems: Through an extensive simulation study, we compare the empirical performance of our method against two tough benchmarks: the projected gradient descent (PGD) method and the Artelys Knitro solver, which is one of the

[4] Vojnovic et al. (2020) actually establish a linear convergence rate guarantee, but with the additional assumption that the negative log-likelihood function is strongly convex. We do not make this assumption, which is generally hard to justify. In the absence of such an assumption, the best possible rate is sub-linear.

most advanced commercial solvers for nonlinear optimization and has been used for estimating choice models in prior literature (Akşin et al. 2013, Dubé et al. 2012). We find that our method obtains significantly lower negative log-likelihood values than the benchmarks for all problem sizes considered. The improvements are particularly large for harder problem instances—those with larger numbers of products (as suggested by our theoretical guarantees) and smaller values of nest dissimilarity parameters. Our method easily scales to large problem sizes with up to 33K products and 38K nodes in the tree. By contrast, the benchmark methods struggle to find good quality solutions even for moderately sized problems (consisting of a few thousand products), despite having substantial computational budgets (average runtime of 2 hours and 5 hours for PGD and Knitro respectively). The Knitro method is particularly poor, failing to complete even a single iteration within the time budget of 5 hours for problems with more than 16K products.

We also demonstrate the robustness of our method on the real-world SUSHI Preference Dataset (Kamishima 2003) consisting of top-10 preference orderings over 100 different sushi varieties submitted by 5000 individuals. We define a natural tree structure leveraging attributes of each sushi variety provided in the dataset. Our analysis on 5M (five million) randomly generated transactions consistent with the preference orderings shows that fitting the more flexible tree logit model achieves better likelihood both in-sample and out-of-sample when compared to the standard MNL model, indicating that there is value to the additional complexity of the tree logit model. The MM method again outperforms both the benchmark PGD and Knitro methods and is robust to the initialization. Interestingly, unlike in the simulation study, we find that the Knitro method outperforms the PGD method, yielding no clear winner between the two benchmark methods.

1.2 Literature Review

The literature on tree logit models is vast and cuts across a range of disciplines. We focus this review on the literature that deals with estimation issues.

The early heuristics for fitting a tree logit model were based on exploiting its connections to the MNL model. The MNL model is a special case of the tree logit model. The likelihood problem associated with the MNL model can be shown to be a convex program, so the MNL parameters can be estimated in a tractable manner and several scalable estimation methods exist (Hunter 2004, Train 2009). Therefore, in early applications, researchers estimated the tree logit model using a "sequential estimation" technique, which decomposed the tree logit into a collection of MNL models; see the discussion in Section 4.2.4 in Train (2009). This method first estimates the parameters at the leaf nodes, *ignoring* all other nodes in the tree. Each leaf node is associated with a product, and each product is associated with an "attraction" parameter. By ignoring the

tree structure, the sequential method first fits an MNL model to the data and uses the estimated MNL parameter for each product as its attraction parameter in the tree logit model. It then estimates the parameters corresponding to the parents of the leaf nodes while holding the attraction parameters constant and ignoring the rest of the nodes in the tree. This estimation step requires fitting another MNL model and results in the nest dissimilarity parameters associated with each of the parents of the leaf nodes. This sequential process is repeated so that in iteration t, the parameters corresponding to the nodes at height[5] t are estimated by holding fixed the parameters of all the nodes of height less than or equal to $t-1$ and ignoring all the nodes at heights $t+1$ or above. The process terminates when it reaches the root node.

The sequential method is computationally fast because each iteration requires fitting only an MNL model and the total number of iterations is equal to the height of the root node. Unfortunately, it is a highly inefficient estimator (in a statistical sense), because it ignores the global structure of the log-likelihood function. In practice, it often produces estimates of poor quality with log-likelihood values far below what is optimal. Daly (1987) compared the sequential estimation method with the method of optimizing the entire log-likelihood function via the Newton-Raphson method and showed that the sequential method performed poorly, and he advocated maximizing the log-likelihood function directly. Brownstone and Small (1989) also showed that the estimates produced by the sequential method can be severely biased.

With the advent of commercial software packages for general-purpose non-linear optimization, researchers have generally shifted to estimating the tree logit parameters by directly maximizing the likelihood function via general-purpose optimization methods. In fact, most commercial packages in use for fitting tree logit models use such generic optimization methods. For example, ALOGIT (`www.alogit.com`) claims to use the Newton-Raphson method as described in Daly (1987); see ALOGIT (2018). The PROC MDC procedure in the SAS software gives the user the choice of three optimization methods for maximizing the log-likelihood function: quasi-Newton, Newton-Raphson, or trust region methods (SAS 2018, p. 1016). Similarly, the NLOGIT subroutine in the STATA software uses general-purpose non-linear optimization algorithms, either Newton-Raphson, Berndt-Hall-Hall-Hausman, Davidon-Fletcher-Powell, or Broyden-Fletcher-Goldfarb-Shanno (STATA 2018a,b). Finally, the NLOGIT extension of LIMDEP software package uses the general-purpose Broyden-Fletcher-Goldfarb-Shanno algorithm for optimizing the log-likelihood function (Greene 2018, p. N-507). See Silberhorn et al. (2008) for an overview of different software packages for estimating the tree logit model, and Nocedal and Wright (2006a) for an overview of various non-linear optimization algorithms.

[5] The height of a node is the length of the longest path from that node to a leaf node.

Although generic optimization methods are better than the sequential method, they often result in sub-optimal likelihood values and run into several numerical issues, such as getting stuck in bad local optima. These issues occur because, as shown by Daganzo and Kusnic (1993) and Mishra et al. (2014), the log-likelihood function is concave in the parameters at the leaf nodes but is **not** concave in the parameters of the non-leaf nodes. This non-concavity, along with the absence of additional structural properties of the log-likelihood function, leads to significant challenges in estimating the parameters using generic optimization methods. For instance, Daganzo and Kusnic (1990) reported discouraging results when maximizing the log-likelihood using the Newton-Raphson method.

It must be noted that the although we focus on the standard specification of the tree logit model, alternative specifications have been investigated (Koppelman and Wen 1998, Hunt 2000, Hensher and Greene 2002), each of which varies in terms of how the utilities are normalized. The specification that is selected is important, because different specifications can result in dramatically different results when applied to the same problem (Koppelman and Wen 1998). Different specifications also provide different types of structure, which can potentially be exploited to ease the computational burden of estimating the parameters. For instance, Li et al. (2015) considered an alternative representation of the tree logit model, in which the log-likelihood function is concave in the parameters of the nodes at each height, when all other parameters are held fixed. They exploited this structure to propose an iterative estimation method that optimizes the log-likelihood function over all the parameters of nodes at a particular height using a generic convex optimization method, while holding all other parameters fixed. Instead of these alternative specifications, we focus on estimating the standard specification of the tree logit model, that is, the one derived by McFadden (1981) from the principle of utility maximization. We do so primarily because the standard specification, unlike the alternative specifications, is consistent with the utility maximization principle by design, and as Koppelman and Wen (1998) noted, it has "intuitively reasonable elasticity relationships and a clear interpretation of utility function parameters."

1.3 Organization and Notation

In the next section, we formally define the tree logit model, provide an alternative and direct proof that the tree logit is a random utility model, and provide an expression for the negative log-likelihood function. In Section 3, we describe the conditions under which the tree logit model is identifiable in terms of the structure of the graph induced by the data. We also demonstrate the non-convexity of the negative log-likelihood objective, which makes the estimation problem challenging. In Section 4, we design a fast iterative method for minimizing the negative log-likelihood function based on the MM principle, by leveraging a novel change-of-variables transformation. We also

establish a convergence guarantee for the MM algorithm and contrast it with that for standard gradient descent. The numerical experiments on synthetic data in Section 5 show that our proposed method scales to large problem instances with complex tree structures and thousands of nodes, outperforming two tough benchmark methods. In addition, using real data, we demonstrate the benefit of fitting a tree logit model compared to the standard MNL model. Finally, in Section 6, we conclude with directions for future research.

Before we proceed, we introduce notation that will be used frequently in the paper. For any scalar $x \in \mathbb{R}$, denote the positive part of x as $(x)^+ := \max(x, 0)$. We denote vectors by bold-lowercase variables such as $\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{\mu}$, etc. Given any $\boldsymbol{x} \in \mathbb{R}^d$, where $\mathbb{R}^d$ is the d-dimensional euclidean space, we define $\{\boldsymbol{x}\}^+ := ((x_j)^+ : j = 1, 2, \ldots d)$. We also use $\|\boldsymbol{x}\|_1$, $\|\boldsymbol{x}\|_2$, and $\|\boldsymbol{x}\|_\infty$ to denote the standard L^1, L^2 and L^∞ norms of $\boldsymbol{x}$. Given two vectors $\boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^d$, we let $\langle \boldsymbol{x}, \boldsymbol{y} \rangle$ denote the standard inner product of $\boldsymbol{x}$ and $\boldsymbol{y}$, i.e. $\langle \boldsymbol{x}, \boldsymbol{y} \rangle = \sum_{j=1}^d x_j \cdot y_j$. For any differentiable function $f(\cdot)$ over $\mathbb{R}^d$, $\nabla f(\bar{\boldsymbol{x}})$ denotes the gradient w.r.t $\boldsymbol{x}$ evaluated at $\bar{\boldsymbol{x}}$, i.e. the vector of partial derivatives $(\partial f(\bar{\boldsymbol{x}})/\partial x_j : j = 1, 2, \ldots, d)$. Finally, for any differentiable function $g : \mathbb{R}^{d_1} \times \mathbb{R}^{d_2} \to \mathbb{R}$, we let $\nabla_{\boldsymbol{x}} g(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ and $\nabla_{\boldsymbol{y}} g(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$ denote respectively the gradients w.r.t $\boldsymbol{x}$ and $\boldsymbol{y}$ evaluated at $(\bar{\boldsymbol{x}}, \bar{\boldsymbol{y}})$.

2. Overview and Problem Formulation

In this section, we provide an overview of the tree logit model and formulate the estimation problem. Let $\mathcal{N} = \{1, \ldots, n\}$ denote the universe of n products, which can include the no-purchase option. We consider a firm that sells products from $\mathcal{N}$. As customers arrive, the firm offers each customer a subset of products to choose from. The customer then either purchases a product from the offered set or leaves without making a purchase, in which case we say that the customer chooses the no-purchase option. Different customers may observe different offer sets, because firms often experience stockouts or make other deliberate adjustments to their assortments. The firm collects purchase transaction data by recording the offer set shown and the corresponding choice made by each arriving customer. The goal is to use these transaction data to describe the observed demand patterns. We consider the case where the choice behavior of the customers is described by a tree logit model and focus on the problem of estimating the model parameters from the historical transaction data. Before we formulate the estimation problem, we describe the tree logit model, the modeling assumptions, and the associated choice probability expressions.

2.1 The Tree Logit Model

The tree logit model assumes that products are organized in a product taxonomy or hierarchy, which is represented as a rooted tree T. We let root denote the root node of the tree, and without

loss of generality, orient the tree with edges directed away from the root node, resulting in a directed out-tree. Each leaf node of the tree corresponds to a product, so the leaf nodes are indexed by $\ell \in \mathcal{N}$. For ease of notation, we use T to denote both the tree and the set of nodes in the tree. The non-leaf nodes are indexed by $j \in \mathsf{T} \setminus \mathcal{N}$. Each non-leaf node j represents a sub-division of the products, specifically, the collection of products in the sub-tree T_j rooted at j. As an example, the product hierarchy in Figure 1 is represented as the tree in Figure 2(a) on page 12. Here, the tree has eight leaf nodes corresponding to the eight itineraries in Figure 1, and the non-leaf node labeled "AM Flights" represents all the morning flight itineraries. For mathematical convenience, we let T_ℓ denote the tree consisting of the single node corresponding to product ℓ, for all $\ell \in \mathcal{N}$.

The model associates a *mean utility* parameter $\mu_\ell \in \mathbb{R}$ with each leaf node (or product) $\ell \in \mathcal{N}$ and a *nest dissimilarity* parameter $\lambda_j \in (0,1]$ with each non-leaf node $j \in \mathsf{T} \setminus \mathcal{N}$. It assumes that in each purchase instance, a customer samples a utility value, $\mathit{utility}_\ell = \mu_\ell + \varepsilon_\ell$, for each product ℓ and then chooses the product with the highest utility; ties are assumed to be broken at random. The parameter μ_ℓ captures the deterministic component of the utility, and the random variable ε_ℓ captures the random component of the product utility. In Section 2.2, we discuss the detailed distributional assumptions on the collection of random variables $(\varepsilon_\ell \colon \ell \in \mathcal{N})$, but for now, we note that the parameter λ_j measures the correlation among the utilities of the products in the sub-tree T_j. In addition to $\lambda_j \in (0,1]$, we impose the constraints that $\lambda_j \le \lambda_{\mathsf{pa}(j)}$ for all $j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})$ and $\lambda_{\mathsf{root}} = 1$, where $\mathsf{pa}(j)$ denotes the *parent* of node j. As shown below, these constraints ensure identification of the nest dissimilarity parameters and that the tree logit model is consistent with the utility maximization principle. For mathematical convenience, we let $\lambda_{\mathsf{pa}(\mathsf{root})} = +\infty$ and $\lambda_\ell = +\infty$ for all leaf nodes $\ell \in \mathcal{N}$.

Defining choice probabilities: The most intuitive way to express the choice probabilities under a tree logit model is in terms of a random walk along the tree, in which the customer traverses the tree hierarchically, starting from root until she reaches a leaf node. In each step of the random walk, the customer moves from the current node to one of its children with probability that is proportional to the "attraction" value of the corresponding child node. For example, when choosing flight itineraries in the example shown in Figure 2(a) on p. 12, a customer may go down the product hierarchy and first decide whether to pick a morning or an afternoon flight, and upon picking, say, a morning flight, may then decide whether to go for a flight with no stops or one stop.

To define the attraction values of the nodes, we start with the case where all the products in the universe are offered. We associate each node j with a weight function $W_j\colon \mathbb{R}^n \times (0,1]^{|\mathsf{T}\setminus\mathcal{N}|} \to \mathbb{R}$ that is defined recursively as follows:

$$W_j(\boldsymbol{\mu},\boldsymbol{\lambda}) = \begin{cases} \mu_j, & \text{if } j \text{ is a leaf node in } \mathsf{T}\ (j\in\mathcal{N}),\\ \lambda_j \log\left(\sum_{k\in\mathsf{Children}(j)} e^{W_k(\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}\right), & \text{if } j \text{ is a non-leaf node in } \mathsf{T}\ (j\in\mathsf{T}\setminus\mathcal{N}). \end{cases}$$

When the parameters $(\boldsymbol{\mu},\boldsymbol{\lambda})$ are clear from the context, we drop the arguments and simply write W_j for each $j\in\mathsf{T}$. The above weight function captures the value to the customer of the entire subset of products T_j. Because the customer eventually chooses only one product, the one with the maximum utility, the value of a subset of products can be thought of as the expected maximum utility value among all the products in T_j, corresponding to $\mathbb{E}[\max_{\ell\in\mathsf{T}_j} \textit{utility}_\ell]$, where the expectation is with respect to the distribution of the collection of random variables $(\varepsilon_\ell\colon \ell\in\mathcal{N})$. The distributional assumptions made by the model lead to the expression of W_j above (see Section 2.2 for more details). Note that it follows immediately from the definition that $W_{\mathsf{pa}(j)} \geq W_j$ for each node $j\in\mathsf{T}\setminus\{\mathsf{root}\}$, which is as expected, because $\mathsf{T}_{\mathsf{pa}(j)} \supseteq \mathsf{T}_j$ and the maximum utility value among the products in $\mathsf{T}_{\mathsf{pa}(j)}$ is therefore greater than or equal to that among the products in T_j.

Now, the probability that a customer chooses product $\ell\in\mathcal{N}$ is equal to the probability that a random walk starting from root ends at node ℓ. Starting from root, the random walk proceeds along the directed edges. In each step, the walk moves from node j to the child node $k\in\mathsf{Children}(j)$ with probability that is proportional to its attraction value, e^{W_k/λ_j}. More precisely, the probability $\psi_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})$ that the walk goes from node $j=\mathsf{pa}(k)$ to node k is given by

$$\psi_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \frac{e^{W_k(\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\mathsf{pa}(k)}}}{\sum_{i\in\mathsf{Children}(\mathsf{pa}(k))} e^{W_i(\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\mathsf{pa}(k)}}} = e^{-(W_{\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})-W_k(\boldsymbol{\mu},\boldsymbol{\lambda}))/\lambda_{\mathsf{pa}(k)}}\ , \tag{1}$$

where the last equality follows from the definition of $W_{\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})$.

More generally, the probability $\psi_{j_1\to j_2}(\boldsymbol{\mu},\boldsymbol{\lambda})$ that the random walk goes from node j_1 to another node j_2 is equal to the product of the probabilities of all the edges that occur along the unique path from j_1 to j_2 in the tree. Let $\mathsf{path}[j_1,j_2]$ denote the collection of nodes in the unique directed path $j_1\to k_1\to k_2\to\cdots k_m\to j_2$ in the tree from node j_1 to j_2, including both j_1 and j_2. If there is no directed path from j_1 to j_2, then $\mathsf{path}[j_1,j_2]=\varnothing$. Further, let $\mathsf{path}(j_1,j_2]$ and $\mathsf{path}[j_1,j_2)$ denote the collections of nodes on the paths $k_1\to k_2\to\cdots k_m\to j_2$ (excluding j_1) and $j_1\to k_1\to k_2\to\cdots k_m$ (excluding j_2), respectively. Similarly, let $\mathsf{path}(j_1,j_2)$ denote the collections of nodes on the paths $k_1\to k_2\to\cdots k_m$ (excluding both j_1 and j_2). Then, the probability $\psi_{j_1\to j_2}(\boldsymbol{\mu},\boldsymbol{\lambda})$ is given by

$$\psi_{j_1\to j_2}(\boldsymbol{\mu},\boldsymbol{\lambda}) \stackrel{\text{def}}{=} \prod_{k\in\mathsf{path}(j_1,j_2]} \psi_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\ ,$$

with the convention that $\psi_{j_1 \to j_2}(\boldsymbol{\mu}, \boldsymbol{\lambda}) = 0$ if $\mathsf{path}(j_1, j_2] = \varnothing$. Using Equation (1), the above expression may be alternatively written as

$$-\log \psi_{j_1 \to j_2}(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \sum_{k \in \mathsf{path}(j_1, j_2]} \frac{W_{\mathsf{pa}(k)}(\boldsymbol{\mu}, \boldsymbol{\lambda}) - W_k(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(k)}} \ . \tag{2}$$

The probability $\mathbb{P}_\ell(\mathcal{N}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ that the customer purchases product ℓ from the universe $\mathcal{N}$ is then equal to $\psi_{\mathsf{root} \to \ell}(\boldsymbol{\mu}, \boldsymbol{\lambda})$.

The choice probability expressions above extend immediately to the case where only a subset $\mathcal{S} \subseteq \mathcal{N}$ of products is offered. The probability is given by a similar random walk probability, but the random walk is now defined on the sub-tree induced by $\mathcal{S}$. More specifically, let $\mathsf{T}[\mathcal{S}]$ denote all the nodes that are ancestors of nodes in $\mathcal{S}$; that is,

$$\mathsf{T}[\mathcal{S}] \stackrel{\text{def}}{=} \{j \in \mathsf{T} \colon \mathsf{path}[j, \ell] \neq \varnothing \text{ for some } \ell \in \mathcal{S}\} \ .$$

Figure 2(b) shows an example of $\mathsf{T}[\mathcal{S}]$. We also use the notation $\mathsf{T}[\mathcal{S}]$ to denote the sub-tree of T induced by the set of nodes $\mathsf{T}[\mathcal{S}]$. Note that the set of leaf nodes of the tree $\mathsf{T}[\mathcal{S}]$ is exactly $\mathcal{S}$, whereas the non-leaf nodes are the ancestors of the nodes in $\mathcal{S}$.

We define the corresponding weight function $W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ at each node $j \in \mathsf{T}[\mathcal{S}]$ as follows:

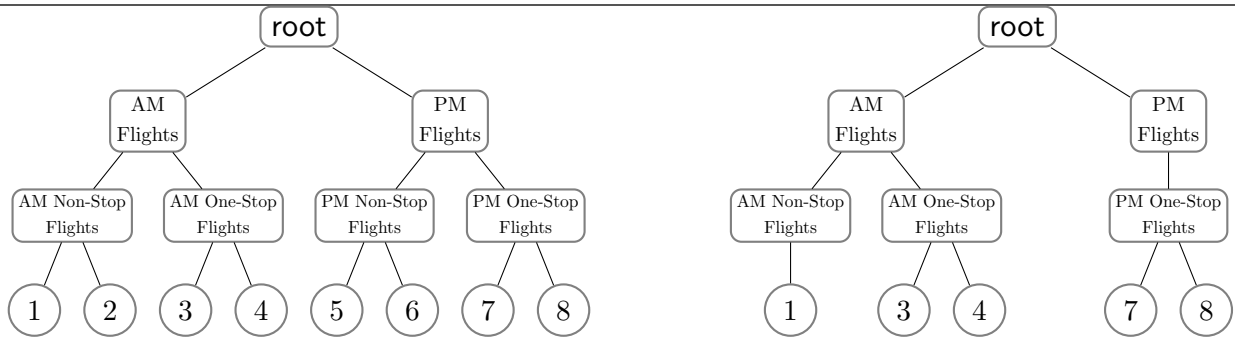
$$W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) = \begin{cases} \mu_j, & \text{if } j \text{ is a leaf node in } \mathsf{T}[\mathcal{S}] \ (j \in \mathcal{S}), \\ \lambda_j \log\left(\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_j}\right), & \text{if } j \text{ is a non-leaf node in } \mathsf{T}[\mathcal{S}] \ (j \in \mathsf{T}[\mathcal{S}] \setminus \mathcal{S}) \ . \end{cases}$$

Note that $\mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]$ is the set of children of node j in the sub-tree $\mathsf{T}[\mathcal{S}]$. As above, we define the random walk probability $\psi_{j_1 \to j_2}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ from j_1 to j_2 under $\mathcal{S}$ by replacing $W_j(\boldsymbol{\mu}, \boldsymbol{\lambda})$ in Equation (1) with $W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$. For each $\ell \in \mathcal{S}$, the probability $\mathbb{P}_\ell(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ that the customer chooses product $\ell \in \mathcal{S}$ is then equal to $\psi_{\mathsf{root} \to \ell}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$.

2.2 Interpretation as a Random Utility Model

We now show how the choice probabilities under a tree logit model can be derived from first principles using its utility specification. In the process, we specify the precise distributional assumptions that the model makes for the random terms $(\varepsilon_\ell \colon \ell \in \mathcal{N})$ in its utility specification and establish that it is a member of the random utility maximization (RUM) class.

The RUM family is a general class of choice models that is designed to capture the distributions of choices observed in a population with varying preferences across customers. In its most general form, it assumes that each customer samples utility values for each of the offered products (including the no-purchase alternative) according to some distribution and then chooses the product with



(a) Original Tree T (b) Sub-tree $\mathsf{T}[\mathcal{S}]$ for $\mathcal{S}=\{1,3,4,7,8\}$

Figure 2 (a) The tree representation of the product hierarchy in Figure 1. There are eight leaf (circle) nodes corresponding to the eight itineraries, indexed by $\mathcal{N}=\{1,2,3,4,5,6,7,8\}$. The non-leaf (rectangle) nodes are "AM Flights", "PM Flights", "AM Non-Stop Flights", "AM One-Stop Flights", "PM Non-Stop Flights", and "PM One-Stop Flights", which correspond to groupings of itineraries. (b) The sub-tree $\mathsf{T}[\mathcal{S}]$ when $\mathcal{S}=\{1,3,4,7,8\}$, which consists of $\mathcal{S}$ at the leaves and all the ancestors of nodes in $\mathcal{S}$. Using $\lambda_{\mathsf{root}}=1$, the probability $\mathbb{P}_4(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})=\psi_{\mathsf{root}\to 4}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ that the customer chooses itinerary $4\in\mathcal{S}$ is given by

$$\begin{aligned}\psi_{\mathsf{root}\to 4}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) &= \psi_{\mathsf{root}\to \mathrm{AM}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})\times\psi_{\mathrm{AM}\to\text{AM One-Stop}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})\times\psi_{\text{AM One-Stop}\to 4}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})\\ &= \frac{e^{W_{\mathrm{AM}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}}{e^{W_{\mathrm{AM}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}+e^{W_{\mathrm{PM}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}}\times\frac{e^{W_{\text{AM One-Stop}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\mathrm{AM}}}}{e^{W_{\text{AM Non-Stop}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\mathrm{AM}}}+e^{W_{\text{AM One-Stop}}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\mathrm{AM}}}}\\ &\qquad\times\frac{e^{W_4(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\text{AM One-Stop}}}}{e^{W_3(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\text{AM One-Stop}}}+e^{W_4(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_{\text{AM One-Stop}}}}\ .\end{aligned}$$

the highest utility. The randomness in the utilities captures the variation of preferences across the population and generalizes the notion of a single customer maximizing her utility to a population of customers with heterogeneous preferences maximizing their respective utilities. Different members of the RUM class differ in terms of the specific distributional assumptions they make for the utilities. For instance, the MNL model assumes that the random components $(\varepsilon_\ell\colon \ell\in\mathcal{N})$ of the utilities are independent and identically distributed Gumbel[6] random variables.

Unlike for the MNL model, most developments in the literature do not specify the distribution of the random components in a tree logit model explicitly. Rather, they specify that distribution indirectly through a generating function. McFadden (1981) proposed a family of RUM models called the Generalized Extreme Value (GEV) family. This family is described through a generating function that is required to satisfy a specific set of properties. McFadden (1981) showed that every generating function that satisfies these properties leads to a discrete choice model from the RUM family, with choice probabilities specified in terms of the partial derivatives of the generating

[6] A random variable x follows a Gumbel distribution with a location parameter μ and a scaling parameter β if $\mathbb{P}\{x\le x\}=e^{-e^{-(x-\mu)/\beta}}\ \forall\ x\in\mathbb{R}$, and we write $x\sim\mathsf{Gumbel}(\mu,\beta)$. Note that in the specification we use, the scaling parameter β appears in the denominator; that is, β divides $(x-\mu)$. In other specifications used in the literature, the scaling parameter appears in the numerator; that is, β multiplies $(x-\mu)$.

function. The tree logit model can be shown to be a member of the GEV family through an appropriate choice of the generating function; see McFadden (1981) and Li et al. (2015, Appendix G). Although the generating function approach does provide a mathematical proof that the tree logit model is a member of the RUM family, it does not provide an intuitive economic justification of the underlying distributional assumptions. Indeed, as noted on page 93 in Train (2009), "there is little economic intuition to motivate these properties" of the generating functions.

Instead of taking the indirect generating function approach, we present below an explicit construction of the random variables $(\varepsilon_\ell : \ell \in \mathcal{N})$, making explicit the underlying correlation structures of the utilities. We decompose each random variable ε_ℓ into a sum of random variables associated with the nodes in the tree T. For each leaf node $\ell \in \mathcal{N}$, let $v_\ell \sim \mathsf{Gumbel}\left(0, \lambda_{\mathsf{pa}(\ell)}\right)$ be a Gumbel random variable with location parameter 0 and scaling parameter $\lambda_{\mathsf{pa}(\ell)}$. For each non-leaf node j such that $j \neq \mathsf{root}$, let v_j be a random variable that is independent of everything else, with the property that $v_j + \mathsf{Gumbel}(0, \lambda_j) \sim \mathsf{Gumbel}(0, \lambda_{\mathsf{pa}(j)})$. Note that such a random variable exists by Corollary A.3 in the Appendix because $\lambda_j \leq \lambda_{\mathsf{pa}(j)}$. For mathematical convenience, we set $v_{\mathsf{root}} = 0$. All the random variables $\{v_j : j \in \mathsf{T}\}$ are independent of each other. Define $\varepsilon_\ell = \sum_{j \in \mathsf{path}(\mathsf{root}, \ell]} v_j$, so

$$\mathit{utility}_\ell \;=\; \mu_\ell + \varepsilon_\ell \;=\; \mu_\ell \;+ \sum_{j \in \mathsf{path}(\mathsf{root}, \ell]} v_j \;.$$

Thus, the random component of the utility of product ℓ is the sum of all the random variables in the unique path from root to ℓ. It is instructive to derive the correlation structure of the random variables $(\varepsilon_\ell : \ell \in \mathcal{N})$. It follows from our definitions that $\mathsf{Var}(v_\ell) = \pi^2 \lambda^2_{\mathsf{pa}(\ell)}/6$ for each leaf node $\ell \in \mathcal{N}$ and that $\mathsf{Var}(v_j) = \pi^2(\lambda^2_{\mathsf{pa}(j)} - \lambda^2_j)/6$ for each non-leaf node $j \in \mathsf{T} \setminus (\mathsf{root} \cup \mathcal{N})$. Because $\varepsilon_\ell = \sum_{j \in \mathsf{path}(\mathsf{root}, \ell]} v_j$, we thus obtain by telescoping that $\mathsf{Var}(\varepsilon_\ell) = \sum_{j \in \mathsf{path}(\mathsf{root}, \ell]} \mathsf{Var}(v_j) = \pi^2 \lambda^2_{\mathsf{root}}/6$. In a similar fashion, we can show that the covariance $\mathsf{Cov}(\varepsilon_\ell, \varepsilon_{\ell'}) = \pi^2(\lambda^2_{\mathsf{root}} - \lambda^2_j)/6$, where j is the common ancestor nearest to both ℓ and ℓ'; that is, $\{\ell, \ell'\} \subseteq \mathsf{T}_j$ but $\{\ell, \ell'\} \nsubseteq \mathsf{T}_k$ for all $k \in \mathsf{Children}(j)$. Putting everything together, we obtain that for $\ell \neq \ell'$, if j is the nearest to both ℓ and ℓ', then

$$\mathsf{Corr}(\varepsilon_\ell, \varepsilon_{\ell'}) = \frac{\mathsf{Cov}(\varepsilon_\ell, \varepsilon_{\ell'})}{\sqrt{\mathsf{Var}(\varepsilon_\ell)\,\mathsf{Var}(\varepsilon_{\ell'})}} \;=\; 1 \;-\; \left(\frac{\lambda_j}{\lambda_{\mathsf{root}}}\right)^2 \;.$$

Therefore, the random terms $(\varepsilon_\ell : \ell \in \mathcal{N})$ are identically distributed, *but* they are not independent of each other. The correlation structure shows how the tree logit model captures similarities among products in the same sub-division. Note that $\lambda_j \leq \lambda_{\mathsf{pa}(j)}$. It thus follows that the utilities of two products become more correlated as the common ancestor moves farther away from the root node, or equivalently, closer to the leaf nodes. Applying this to the example shown in Figure 1, we observe that the correlation in the utilities between two morning non-stop flights (Itineraries #1 and #2) is higher than that between a morning and an evening flight.

The following theorem shows that under the RUM principle, the above utility specification gives rise to the same choice probability $\psi_{\mathsf{root}\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ defined in Section 2.1.

Theorem 2.1 (Selection Probability under Random Utility Maximization) *For each subset $\mathcal{S}\subseteq\mathcal{N}$ and for each product $\ell\in\mathcal{S}$,*

$$\mathbb{P}\left\{\mathit{utility}_\ell > \max_{j\in\mathcal{S}\setminus\{\ell\}} \mathit{utility}_j\right\} \;=\; \psi_{\mathsf{root}\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \;=\; \mathbb{P}_\ell\left(\mathcal{S}\,;\,\boldsymbol{\mu},\boldsymbol{\lambda}\right).$$

We provide a self-contained proof of this theorem in Appendix A. Although this result is already known, the proof is scattered across several old papers, see e.g., Zachary (1978), Hunt (2000), some of which are not easily accessible. Moreover, some of the proofs in the literature are not complete, and combining all the known results still requires filling several technical gaps. To the best of our knowledge, our proof is the first complete end-to-end proof for this result.

2.3 The Parameter Estimation Problem

We now formulate the parameter estimation problem under the tree logit model. Assume that the firm has the dataset $\{(\mathcal{S}^q,c^q):q=1,\ldots,Q\}$ consisting of Q transactions, with $\mathcal{S}^q\subseteq\mathcal{N}$ denoting the subset of products offered to the q^{th} customer and $c^q\in\mathcal{S}^q$ denoting the customer's selection, which may be the no-purchase option. We want to use the transaction data to estimate the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ of the model, when we are given the tree structure T. We adopt the maximum likelihood estimation technique, which finds the parameter values that maximize the data log-likelihood value or, equivalently, minimize the negative data log-likelihood value. For each node $j\in\mathsf{T}$ and transaction $q=1,\ldots,Q$, let $\mathsf{T}^q=\mathsf{T}[\mathcal{S}^q]$, $W_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})=W_j(\mathcal{S}^q;\boldsymbol{\mu},\boldsymbol{\lambda})$, and $\psi_{j\to k}^q(\boldsymbol{\mu},\boldsymbol{\lambda})=\psi_{j\to k}(\mathcal{S}^q;\boldsymbol{\mu},\boldsymbol{\lambda})$. The following theorem provides two equivalent expressions for the *negative* data log-likelihood function – or more simply, the negative log-likelihood function – when customers are drawn independently from a population whose preferences are described by the tree logit model.

Theorem 2.2 (Negative Log-Likelihood) *Given the parameters $(\boldsymbol{\mu},\boldsymbol{\lambda})$, the negative data log-likelihood value is given by*

$$\begin{aligned}\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda}) &= \frac{1}{Q}\sum_{q=1}^{Q}\left\{W_{\mathsf{root}}^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \;+\; \sum_{j\in\mathsf{path}(\mathsf{root},c^q)} W_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \;-\; \frac{\mu_{c^q}}{\lambda_{\mathsf{pa}(c^q)}}\right\}\\ &= \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\frac{W_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_j} \;-\; \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\frac{W_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}}\,.\end{aligned}$$

Proof: The negative log-likelihood value of the data is equal to $\frac{1}{Q}\sum_{q=1}^{Q} -\log \psi^q_{\mathsf{root}\to c^q}(\boldsymbol{\mu},\boldsymbol{\lambda})$. The expression in the statement of the lemma is obtained by invoking the expression in Equation (2) for $-\log \psi^q_{\mathsf{root}\to c^q}$ and then re-arranging the terms. More precisely, we have

$$\begin{aligned}\sum_{q=1}^{Q} -\log \psi^q_{\mathsf{root}\to c^q}(\boldsymbol{\mu},\boldsymbol{\lambda}) &= \sum_{q=1}^{Q}\sum_{j\in\mathsf{path}(\mathsf{root},c^q]} \frac{W^q_{\mathsf{pa}(j)}(\boldsymbol{\mu},\boldsymbol{\lambda}) - W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}} \\ &= \sum_{q=1}^{Q}\left\{ W^q_{\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}) + \sum_{j\in\mathsf{path}(\mathsf{root},c^q)} W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})\left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) - \frac{\mu_{c^q}}{\lambda_{\mathsf{pa}(c^q)}}\right\},\end{aligned}$$

where the last equality follows from a straightforward re-arrangement of the terms of the inner summation. A slightly different re-arrangement yields the following expression:

$$\begin{aligned}\sum_{q=1}^{Q} -\log \psi^q_{\mathsf{root}\to c^q}(\boldsymbol{\mu},\boldsymbol{\lambda}) &= \sum_{q=1}^{Q}\sum_{k\in\mathsf{path}(\mathsf{root},c^q]} \frac{W^q_{\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda}) - W^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(k)}} \\ &\overset{\text{(a)}}{=} \sum_{q=1}^{Q}\sum_{k\in\mathsf{T}\setminus\{\mathsf{root}\}} \mathbb{1}_{\{c^q\in\mathsf{T}_k\}}\frac{W^q_{\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(k)}} - \sum_{q=1}^{Q}\sum_{k\in\mathsf{T}\setminus\{\mathsf{root}\}} \mathbb{1}_{\{c^q\in\mathsf{T}_k\}}\frac{W^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(k)}}, \\ &\overset{\text{(b)}}{=} \sum_{q=1}^{Q}\sum_{j\in\mathsf{T}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_j} - \sum_{q=1}^{Q}\sum_{j\in\mathsf{T}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}},\end{aligned}$$

where the equality (a) follows because $k\in\mathsf{path}(\mathsf{root},c^q]$ if and only if c^q belongs to the sub-tree rooted at k; that is, $c^q\in\mathsf{T}_k$. The first term in the last equality (b) follows from the change of variable $j=\mathsf{pa}(k)$. With this change of variable, as k is varied over the set of nodes $\mathsf{T}\setminus\{\mathsf{root}\}$, the variable j varies over the set $\mathsf{T}\setminus\mathcal{N}$. Because $\lambda_\ell=+\infty$ for each leaf node $\ell\in\mathcal{N}$, we can extend the summation to cover the entire set of nodes in T, resulting in the first term. The second term is obtained by replacing k with j, and including root in the summation, since $\lambda_{\mathsf{pa}(\mathsf{root})}=+\infty$. ■

Our goal is to find the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$ that minimize $\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})$. However, the model parameters are *not* identified from the choices alone, unless we normalize the location and the scale of the model. This is the case because as far as choices are concerned, the actual values of utilities do not matter; only the preferences induced by the utilities matter. If all of these utility values are scaled by the same positive constant or shifted by the same value, the induced preference ordering, and hence the choices, remains the same. There are many ways to fix the mean utility parameters. We adopt the common approach of setting the mean utility of a particular product to zero, we set $\mu_1=0$. We will also fix the scale by setting the variance of the random variables $(\varepsilon_\ell:\ell\in\mathcal{N})$ to be a constant, say $\lambda_{\mathsf{root}}=1$. We thus obtain the domains

$$\begin{aligned}\mathcal{D}_1 &= \{\,\boldsymbol{\mu}\in\mathbb{R}^n : \mu_1=0\,\} \qquad \text{and} \\ \mathcal{D}_2 &= \{\boldsymbol{\lambda}\in(0,1]^{|\mathsf{T}\setminus\mathcal{N}|} : \lambda_j\le\lambda_{\mathsf{pa}(j)}\ \forall\ j\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\}),\ \lambda_{\mathsf{root}}=1\},\end{aligned}$$

and the maximum likelihood estimation (MLE) corresponds to the following optimization problem:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\lambda})\in\mathcal{D}_1\times\mathcal{D}_2} \mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda}). \qquad \text{(MLE problem)}$$

In Section 3, we derive conditions under which the parameters of the tree logit model can be identified from the data, and in Section 4 we discuss how to solve the MLE problem.

3. Conditions for Identifiability

We now examine properties of the negative log-likelihood function in the MLE problem to establish conditions for the identification of the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}$. To ensure identification, we need sufficient variation in the observed offer sets and corresponding choices in our data. For instance, we need all the products to be offered at least once; otherwise, we cannot identify the mean parameters of the products that are never offered. We need each product to be purchased at least once; otherwise, the optimal solution to the MLE problem will be unbounded, with the mean utility parameter of the product that is never purchased diverging to $-\infty$. We also need each non-leaf node to be non-degenerate (to have at least two children) in the induced sub-tree T^q for some q; otherwise, the nest dissimilarity parameter corresponding to this non-leaf node cannot be identified. To state our conditions, we need the concept of a *comparison graph*, defined as follows.

Definition 3.1 (Comparison Graph) *Given transaction data $\{(S^q, c^q) : q = 1, \ldots, Q\}$, a comparison graph $\mathsf{Comp} = (\mathcal{N}, \mathsf{E})$ is a directed graph whose nodes correspond to products, and there is a directed edge $(\ell_1, \ell_2) \in \mathsf{E}$ from ℓ_1 to ℓ_2 if there is an offer set $\mathcal{S}^q$ such that $\{\ell_1, \ell_2\} \subseteq \mathcal{S}^q$ and $c^q = \ell_1$.*

The next theorem shows that for each $\boldsymbol{\lambda} \in \mathcal{D}_2$, the strong connectivity of the comparison graph is necessary and sufficient to identify the mean utilities. The proof is given in Appendix D.1.

Theorem 3.2 (Identifying the Mean Utilities) *For each $\boldsymbol{\lambda} \in \mathcal{D}_2$, the optimization problem $\min_{\boldsymbol{\mu}\in\mathcal{D}_1} \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda})$ admits a unique and bounded solution if and only if the comparison graph Comp is strongly connected; that is, there is a directed path between every pair of nodes.*

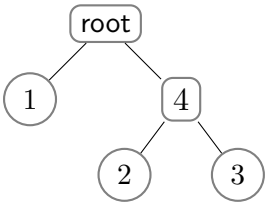
The next result identifies a necessary and sufficient condition for the negative log-likelihood function to be dependent on λ_j. The proof is given in Appendix D.2.

Theorem 3.3 (Identifying the Dissimilarity Parameters) *For each non-leaf node $j \in \mathsf{T} \setminus \mathcal{N}$ such that $j \neq \mathsf{root}$, the function NegLog function varies with respect to λ_j if and only if there exists a transaction $q \in \{1, \ldots, Q\}$ such that node j has at least two children in sub-tree T^q.*

Several remarks are in order here. First, checking whether a graph is strongly connected is a standard problem in computer science, and it can solved efficiently in linear time (Tarjan 1972). Therefore, the condition can be easily checked in practice. Second, the notion of a comparison graph was introduced by Hunter (2004) to show that strong connectedness of the comparison graph is both necessary and sufficient to ensure identification of the parameters of an MNL model. Our result extends that of Hunter (2004) to a tree logit model. Third, note that strong connectedness of the comparison graph immediately implies that each product is purchased at least once in the data; otherwise, the product will have only incoming edges in the comparison graph.

We now focus on solving the MLE problem. The main challenge is that the negative log-likelihood function is non-convex in $\boldsymbol{\lambda}$ as shown in the following example.

Example 3.4 (Non-convexity of NegLog in $\boldsymbol{\lambda}$)



Suppose $\mathcal{N}=\{1,2,3\}$ and the tree structure is shown in the figure to the left, where root has two children: a leaf node 1 and a non-leaf node 4. Node 4 has two children: leaf nodes 2 and 3 . The parameters of the model consist of μ_1, μ_2, μ_3, and λ_4. Suppose we offer the full assortment, and for each $\ell \in \mathcal{N}$, let $s_\ell \in \mathbb{Z}_{++}$ denote the number of customers who select product ℓ. Then,

$$\begin{aligned}
\psi_{\mathsf{root}\to 1}(\boldsymbol{\mu},\lambda_4) &= \psi_{\mathsf{root}\to 1}(\boldsymbol{\mu},\lambda_4) &&= \frac{e^{\mu_1}}{e^{\mu_1}+\left[e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}\right]^{\lambda_4}} \\
\psi_{\mathsf{root}\to 2}(\boldsymbol{\mu},\lambda_4) &= \psi_{\mathsf{root}\to 4}(\boldsymbol{\mu},\lambda_4)\times\psi_{4\to 2}(\boldsymbol{\mu},\lambda_4) &&= \frac{\left[e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}\right]^{\lambda_4}}{e^{\mu_1}+\left[e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}\right]^{\lambda_4}}\times\frac{e^{\mu_2/\lambda_4}}{e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}} \\
\psi_{\mathsf{root}\to 3}(\boldsymbol{\mu},\lambda_4) &= \psi_{\mathsf{root}\to 4}(\boldsymbol{\mu},\lambda_4)\times\psi_{4\to 3}(\boldsymbol{\mu},\lambda_4) &&= \frac{\left[e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}\right]^{\lambda_4}}{e^{\mu_1}+\left[e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}\right]^{\lambda_4}}\times\frac{e^{\mu_3/\lambda_4}}{e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}}\ ,
\end{aligned}$$

which implies that

$$\begin{aligned}
\mathsf{NegLog}(\boldsymbol{\mu},\lambda_4) \;=\; \Big((s_1+s_2+s_3)\log\Big(e^{\mu_1}+\left[e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}\right]^{\lambda_4}\Big) \;+\; (s_2+s_3)(1-\lambda_4)\log\left(e^{\mu_2/\lambda_4}+e^{\mu_3/\lambda_4}\right) \\
-\; s_1\mu_1 \;-\; s_2\frac{\mu_2}{\lambda_4} \;-\; s_3\frac{\mu_3}{\lambda_4}\Big) \Big/ (s_1+s_2+s_3)\ .
\end{aligned}$$

Suppose we have five customers, with $(s_1,s_2,s_3)=(1,1,3)$. For $\boldsymbol{\mu}=(\mu_1,\mu_2,\mu_3)=(0,1,1.03)$, we have $\mathsf{NegLog}(\boldsymbol{\mu},\lambda_4=0.1)=1.0116$ and $\mathsf{NegLog}(\boldsymbol{\mu},\lambda_4=0.3)=1.0381$, and we have that

$$\mathsf{NegLog}(\boldsymbol{\mu},\lambda_4=0.2)=1.0317 \;>\; \frac{\mathsf{NegLog}(\boldsymbol{\mu},\lambda_4=0.1)\;+\;\mathsf{NegLog}(\boldsymbol{\mu},\lambda_4=0.3)}{2} \;=\; 1.0249\ ,$$

which shows that $\lambda_4 \mapsto \mathsf{NegLog}(\boldsymbol{\mu},\lambda_4)$ is not convex in λ_4.

4. MM method for solving the MLE problem

Example 3.4 sheds light on the non-convexity of the negative log-likelihood function, which generally rules out direct optimization with an off-the-shelf solver because of the numerical stability issues reported in prior literature, as discussed in Section 1. Moreover, general purpose solvers do not scale very well to large-scale settings with hundreds or even thousands of products that are of interest to us. Therefore, we consider first-order methods which only require the objective function value and the corresponding gradient vector at each estimate and are known to scale well. While there are many variations, first-order methods generally involve computing the gradient of the objective function at each estimate, and then taking a step of an appropriate size in the direction of the gradient to obtain the new estimate. When the problem is constrained, the new estimate is projected back to the feasible region.

We exploit the tree structure of the model to show that the gradient of the negative log-likelihood function can be computed efficiently, even in large-scale settings, recursively over the tree, starting at the root node and then moving to the leaf nodes (see Appendix C). The challenge in implementing a first-order method then lies in (1) choosing a good step size, and (2) computing the projection. In particular, given a step size α, the projection step requires solving the following problem, given an estimate $(\boldsymbol{\mu}^{(s)}, \boldsymbol{\lambda}^{(s)})$:

$$(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\lambda}^{(s+1)}) \leftarrow \underset{(\boldsymbol{\mu},\boldsymbol{\lambda})\in\mathcal{D}_1\times\mathcal{D}_2}{\arg\min} \left\| (\boldsymbol{\mu}, \boldsymbol{\lambda}) - \left((\boldsymbol{\mu}^{(s)}, \boldsymbol{\lambda}^{(s)}) - \alpha \cdot \nabla \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\lambda}^{(s)}) \right) \right\|_2^2 \tag{3}$$

Recall that the feasible region $\mathcal{D}_2$ for the non-leaf node parameters involves many constraints on $\boldsymbol{\lambda}$ and consequently, the above projection step does not admit a closed-form solution, forcing us to rely on numerical methods. Moreover, the projection step is typically embedded in an outer procedure for choosing the "best" step size, requiring us to solve the above problem repeatedly for different candidate step sizes. As a result, the procedure becomes very slow in practice, especially for large problem instances as observed in our numerical experiments in Section 5.

We design our estimation method to overcome these two challenges. There are two key ingredients in our approach. First, instead of working in the $(\boldsymbol{\mu}, \boldsymbol{\lambda}) \in \mathcal{D}_1 \times \mathcal{D}_2$ domain, we propose a variable transformation and work instead in the $(\boldsymbol{\mu}, \boldsymbol{\delta}) \in \mathcal{D}_1 \times \mathbb{R}_+^{|\mathsf{T}\setminus\mathcal{N}'|}$ domain (the precise mapping is defined in Section 4.1 below). With this change-of-variables, the negative log-likelihood becomes convex in $\boldsymbol{\mu}$ for any fixed $\boldsymbol{\delta}$, and a difference of strictly convex functions (DSC) in $\boldsymbol{\delta}$ for any fixed $\boldsymbol{\mu}$. We then design an *alternating minimization* procedure, which is an iterative algorithm that alternates between updating $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ while keeping the other variable fixed. Second, we exploit the convexity of the negative log-likelihood in $\boldsymbol{\mu}$ and the DSC representation in $\boldsymbol{\delta}$ to develop a majorize-minimize

(MM) method that provides closed-form expressions for updating the estimates of $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ in each iteration, and guarantees an improving solution. In particular, our algorithm does not require any step size tuning unlike typical gradient-based methods, resulting in an overall procedure that converges very fast in practice.

We provide more details on each of the two ingredients below. For the rest of the paper, we assume that the identification conditions of Theorems 3.2 and 3.3 are satisfied.

4.1 Variable transformation

Instead of working in the $(\boldsymbol{\mu}, \boldsymbol{\lambda})$-domain, we use the $(\boldsymbol{\mu}, \boldsymbol{\delta}) \in \mathbb{R}^n \times \mathbb{R}_+^{|\mathsf{T}\setminus\mathcal{N}|}$ domain with the following change of variables:

$$\delta_j := \log \lambda_{\mathsf{pa}(j)} \; - \; \log \lambda_j \quad \text{for all } j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\}) \qquad \text{and} \qquad \delta_{\mathsf{root}} = 0.$$

It is easy to see that there is a one-to-one correspondence between $\boldsymbol{\lambda}$ and $\boldsymbol{\delta}$. In particular, it can be verified that $\lambda_j = e^{-\sum_{k\in\mathsf{path}[\mathsf{root},j]}\delta_k}$ for all $j \in \mathsf{T}\setminus\mathcal{N}$. Note that when $j = \mathsf{root}$, we have $e^{-\sum_{k\in\mathsf{path}[\mathsf{root},\mathsf{root}]}\delta_k} = e^{-\delta_{\mathsf{root}}} = 1 = \lambda_{\mathsf{root}}$. Moreover, we must have $\delta_j \geq 0$ for all $j \in \mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})$ because $\lambda_j \leq \lambda_{\mathsf{pa}(j)}$. To make our notation compact, given any $\boldsymbol{\delta} \in \mathbb{R}_+^{|\mathsf{T}\setminus\mathcal{N}|}$, we let $\delta_{[j]}$ denote the sum $\sum_{k\in\mathsf{path}[\mathsf{root},j]}\delta_k$, for all $j \in \mathsf{T}\setminus\mathcal{N}$. With this notation, it follows that $\lambda_j = \exp\left(-\delta_{[j]}\right)$ for all $j \in \mathsf{T}\setminus\mathcal{N}$. Finally, we use $\boldsymbol{\lambda}(\boldsymbol{\delta})$ to denote the vector $\left(\exp\left(-\delta_{[j]}\right) : j \in \mathsf{T}\setminus\mathcal{N}\right)$.

We overload notation and write $\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$ to denote the negative log-likelihood function under the transformed variables. We also define $\mathsf{Dom}_1 = \{\boldsymbol{\mu} \in \mathbb{R}^n : \mu_1 = 0\}$ and $\mathsf{Dom}_2 = \left\{\boldsymbol{\delta} \in \mathbb{R}_+^{|\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})|}\right\}$. The MLE problem problem can then be written as follows:

$$\min_{(\boldsymbol{\mu},\boldsymbol{\delta})\in\mathsf{Dom}_1\times\mathsf{Dom}_2} \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$$

The following theorem establishes an important property of the NegLog function under the variable transformation. The proof of this result is given in Appendix E.1.

Theorem 4.1 (Structure of NegLog in the transformed space) *For each $\boldsymbol{\delta} \in \mathsf{Dom}_2$, the mapping $\boldsymbol{\mu} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$ is strictly convex on Dom_1. Moreover, for each $\boldsymbol{\mu} \in \mathsf{Dom}_1$, the function $\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$ can be written as a difference of two strictly convex functions (DSC); that is,*

$$\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}) \; = \; \underbrace{\frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} W_j^q\left(\boldsymbol{\mu}, \boldsymbol{\lambda}(\boldsymbol{\delta})\right)\cdot e^{\delta_{[j]}}}_{F_1(\boldsymbol{\mu},\boldsymbol{\delta})} - \underbrace{\frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\{\mathsf{root}\}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} W_j^q\left(\boldsymbol{\mu}, \boldsymbol{\lambda}(\boldsymbol{\delta})\right)\cdot e^{\delta_{[\mathsf{pa}(j)]}}}_{F_2(\boldsymbol{\mu},\boldsymbol{\delta})},$$

where mappings $\boldsymbol{\delta} \mapsto F_1(\boldsymbol{\mu}, \boldsymbol{\delta})$ and $\boldsymbol{\delta} \mapsto F_2(\boldsymbol{\mu}, \boldsymbol{\delta})$ are strictly convex on Dom_2.

As alluded to before, we utilize the above representation of the negative log-likelihood function to design an alternating minimization procedure for solving the MLE problem.

4.2 Alternating minimization for finding a stationary point

Because the MLE problem is non-convex in the parameters $(\boldsymbol{\mu}, \boldsymbol{\delta})$, it is difficult to guarantee convergence to a global minimum in general. Instead, it is common to show convergence to a stationary point of the problem, defined as follows:

Definition 4.2 (Stationary Points) A vector $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) \in \mathsf{Dom}_1 \times \mathsf{Dom}_2$ is a stationary point of the MLE problem if

(a) $\partial \mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}})/\partial \mu_\ell = 0$, for each $\ell \in \mathcal{N} \setminus \{1\}$.
(b) $\left\langle \nabla_{\boldsymbol{\delta}} \mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}), \boldsymbol{\delta} - \bar{\boldsymbol{\delta}} \right\rangle \geq 0$ for all $\boldsymbol{\delta} \in \mathsf{Dom}_2$.

Note that condition (b) is standard for a constrained optimization problem, and states that there is no "descent direction" from the point $\bar{\boldsymbol{\delta}}$, that is, no (feasible) direction provides an improving solution; see, e.g., Lacoste-Julien (2016). Because the domain of the optimization problem is a Cartesian product of Dom_1 and Dom_2, the solution $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}})$ is a stationary point of the MLE problem if $\bar{\boldsymbol{\mu}}$ is a stationary point of the optimization problem $\min_{\boldsymbol{\mu} \in \mathsf{Dom}_1} \mathsf{NegLog}(\boldsymbol{\mu}, \bar{\boldsymbol{\delta}})$ and $\bar{\boldsymbol{\delta}}$ is a stationary point of the optimization problem $\min_{\boldsymbol{\delta} \in \mathsf{Dom}_2} \mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta})$.

The MLE problem lends itself to an *alternating minimization* approach, where, as the name suggests, we alternately optimize the negative log-likelihood objective over $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$ while holding the other fixed. More formally, starting from some initial solution $(\boldsymbol{\mu}^{(0)}, \boldsymbol{\delta}^{(0)})$, the algorithm generates a sequence of iterates $(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})_{s \geq 1}$ by solving the following two subproblems in each iteration s:

$$\boldsymbol{\mu}^{(s+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{\mu} \in \mathsf{Dom}_1} \ \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}^{(s)}) \qquad (\mu \text{ update-subproblem})$$

$$\boldsymbol{\delta}^{(s+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{\delta} \in \mathsf{Dom}_2} \ \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}) \qquad (\delta \text{ update-subproblem})$$

In principle, any standard non-linear solver can be used to solve the above two subproblems. However, we exploit the structure of the negative log-likelihood objective and the constraint regions to design an efficient procedure for solving the μ update-subproblem and δ update-subproblem that results in closed-form updates. Our algorithm is based on the majorization-minimization (MM) framework (Hunter and Lange 2004). We present below the pseudo-code of our algorithm, which we refer to as A-MM, and defer the details to Section 4.3.

Initialization: Pick an initial starting point $\left(\boldsymbol{\mu}^{(0)}, \boldsymbol{\delta}^{(0)}\right) \in \mathsf{Dom}_1 \times \mathsf{Dom}_2$. Denote $\mathsf{sales}_\ell = \sum_{q=1}^{Q} \mathbb{1}_{\{c^q = \ell\}} / Q$ for each $\ell \in \mathcal{N}$.

Description: For each iteration $s = 0, 1, 2, \ldots$, complete the following two steps:

Step 1: Update the mean utility parameters: For each $\ell \in \mathcal{N}$, update $\mu_\ell^{(s+1)}$ as:

$$\tilde{\mu}_\ell^{(s+1)} = \mu_\ell^{(s)} + \exp\left(-\delta_{[\mathsf{pa}(\ell)]}^{(s)}\right) \cdot \log\left(\frac{\mathsf{sales}_\ell}{\mathsf{sales}_\ell + \exp\left(-\delta_{[\mathsf{pa}(\ell)]}^{(s)}\right) \cdot \partial \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})/\partial \mu_\ell}\right)$$

$$\mu_\ell^{(s+1)} = \tilde{\mu}_\ell^{(s+1)} - \tilde{\mu}_1^{(s+1)}$$

Step 2: Update the nest dissimilarity parameters: For each $j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})$, update $\delta_j^{(s+1)}$ as:

$$\delta_j^{(s+1)} = \left(\delta_j^{(s)} - \alpha^{(s)} \cdot \partial \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)})/\partial \delta_j\right)^+,$$

where $\alpha^{(s)}$ is the solution of the following linesearch problem (which can be solved in closed form as shown in Theorem 4.3 below),

$$\underset{\alpha \in \mathbb{R}_+}{\arg\min} \quad H^{(s)}\left(\left\{\boldsymbol{\delta}^{(s)} - \alpha \cdot \nabla_{\boldsymbol{\delta}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)})\right\}^+\right),$$

with the function $H^{(s)} \colon \mathsf{Dom}_2 \to \mathbb{R}_+$ defined as

$$H^{(s)}(\boldsymbol{\delta}) = F_1(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}) - F_2(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \sum_{j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})} \frac{\partial F_2}{\partial \delta_j}\left(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)}\right) \cdot \left(\delta_j - \delta_j^{(s)}\right)$$

In the description above, the functions $F_1(\boldsymbol{\mu}, \boldsymbol{\delta})$ and $F_2(\boldsymbol{\mu}, \boldsymbol{\delta})$ are as defined in Theorem 4.1. Note that the algorithm requires access to the derivatives of the negative log-likelihood objective, which can be computed efficiently as shown in Lemma C.2 and Lemma E.1 in the appendix. Now, it is clear that Step 1 above can be performed efficiently. Moreover, the following theorem establishes that $\alpha^{(s)}$ in Step 2 can be computed in closed form, so that the A-MM algorithm performs closed-form updates in each iteration:

Theorem 4.3 (Step 2 of A-MM algorithm can be solved in closed form) *For each $s \geq 0$, the mapping $\alpha \mapsto H^{(s)}\left(\left\{\boldsymbol{\delta}^{(s)} - \alpha \cdot \nabla_{\boldsymbol{\delta}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)})\right\}^+\right)$ is piecewise convex on $\mathbb{R}_+$ with at most $|\mathsf{T} \setminus \mathcal{N}|$ pieces. Moreover, each piece is either constant or strictly convex, and this can be exploited to solve the linesearch in Step 2 of the A-MM algorithm in closed form.*

The proof of the result is given in Appendix E.2 and leverages the fact that the minimizer of a one-dimensional strictly convex function over an interval can be computed very efficiently, using many well-known algorithms such as the golden-section search (Kiefer 1953).

We now derive the convergence rate of our proposed A-MM algorithm. The following theorem establishes the improvement in the negative log-likelihood objective guaranteed when updating the mean utility parameters according to Step 1 of the A-MM algorithm:

Theorem 4.4 (Improvement via MM update for $\boldsymbol{\mu}$) *Suppose there exists a $\delta_{\mathsf{upper}} > 0$ such that $\delta_j^{(s)} \leq \delta_{\mathsf{upper}}$ for all $j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})$. Then, Step 1 of the* A-MM *algorithm guarantees the following improvement in the negative log-likelihood objective:*

$$\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \leq -\frac{\lambda_{\mathsf{lower}}^2}{2} \left\| \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_1^2$$

where $\lambda_{\mathsf{lower}} = e^{-(\mathsf{height}(\mathsf{root})-1)\times\delta_{\mathsf{upper}}}$, and $\mathsf{height}(\mathsf{root})$ is the height of the root node in T.

We call the constant above λ_{lower} since an upper bound on $\boldsymbol{\delta}$ implies a lower bound on $\boldsymbol{\lambda}$. The proof of the result is given in Appendix E.3 and uses the specific form of the majorizing surrogate presented in Theorem 4.8 later. In particular, the above result implies that Step 1 of the A-MM algorithm generates an improving solution, i.e. a solution with a strictly lower negative log-likelihood objective, as long as the current solution is not a stationary point according to Definition 4.2.

The condition that the $\boldsymbol{\delta}$ estimate is bounded above is similar to the standard smoothness assumption on the objective function when proving convergence rate guarantees (Bubeck 2015). In particular, if $\delta_j^{(s)}$ diverges for some $j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})$, it can be verified from Theorem 2.2 that the NegLog function is no longer smooth, which makes it challenging to establish a bound on the improvement in the objective value in iteration s. However, we point out that the assumption is needed only to establish the improvement bound, the A-MM algorithm can itself be implemented without the knowledge of δ_{upper}.

It is instructive to contrast the improvement guarantee in Theorem 4.4 above to that obtained when using gradient descent to update the estimates in each iteration. By leveraging standard arguments in the literature, the following can be established:

Theorem 4.5 (Improvement via GD update for $\boldsymbol{\mu}$) *Suppose there exists a $\delta_{\mathsf{upper}} > 0$ such that $\delta_j^{(s)} \leq \delta_{\mathsf{upper}}$ for all $j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})$. Let $\boldsymbol{\mu}_{\mathsf{GD}}^{(s+1)}$ denote the gradient descent (GD) update with step-size $1/L$, where $L > 0$ is the smoothness constant of the mapping $\boldsymbol{\mu} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}^{(s)})$. Then, the GD update guarantees the following improvement in the negative log-likelihood objective:*

$$\begin{aligned}\mathsf{NegLog}(\boldsymbol{\mu}_{\mathsf{GD}}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) &\leq -\frac{1}{2L} \left\| \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_2^2 \\ &\leq -\frac{1}{2Ln} \left\| \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_1^2\end{aligned}$$

Moreover, the smoothness constant L is bounded above by $1/\lambda_{\mathsf{lower}}^2$ and this bound is tight up to constant factors.

The first inequality in the result above is the standard improvement guarantee for gradient descent on smooth objective functions. The second inequality follows from noting that $\left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s)},\boldsymbol{\delta}^{(s)})\right\|_2 \geq \frac{1}{\sqrt{n}}\left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s)},\boldsymbol{\delta}^{(s)})\right\|_1$, where recall that n is the dimension of the gradient vector. Now, whether the upper bound in Theorem 4.4 is larger than that in Theorem 4.5 depends on how $L\cdot\lambda^2_{\mathsf{lower}}$ scales with the number of products n. The proof in Appendix E.4 shows that $L \leq \frac{1}{\lambda^2_{\mathsf{lower}}}$ and this bound is tight (up to constant factors), so that $L\cdot\lambda^2_{\mathsf{lower}}$ does not depend on n in the worst-case. Consequently, our proposed MM updates can generate significantly better objective values for the same computational budget compared to GD updates, especially when the number of products is large. Indeed, our numerical results in Section 5.1 confirm this insight. Moreover, the improvement bound for the GD update assumes that the step size can be chosen appropriately—either by computing the smoothness constant or via linesearch—which can increase the computational burden and further slow down convergence, especially on large problem instances. Our A-MM algorithm, on the other hand, is favorable in this regard as it provides a closed-form update without the need for any step size tuning.

We can also bound the improvement in the negative log-likelihood objective guaranteed by Step 2 of the A-MM algorithm. The improvement bound, shown in Lemma F.1 in Appendix F, follows from standard arguments in the literature and is similar to that obtained when using (projected) GD to update the $\boldsymbol{\delta}$ estimate. Finally, we can combine Theorem 4.4 and Lemma F.1 to establish a sublinear rate of convergence of the A-MM algorithm to a stationary point of the MLE problem, see Theorem G.1 in Appendix G.

4.2.1 Convergence guarantee for the MNL model. As detailed below after the statement of Theorem 4.8, the A-MM algorithm reduces to the standard MM algorithm for estimating the parameters of an MNL model. Because the MNL model has more structure ($\mathsf{T}=\mathcal{N}\cup\{\mathsf{root}\}$) and the negative log-likelihood function is convex in the mean utility parameters $\boldsymbol{\mu}$, we can establish stronger convergence rate guarantees using the improvement bounds from Theorems 4.4 and 4.5:

Theorem 4.6 (Faster convergence of MM compared to GD for the MNL model) *Suppose that* $\mathsf{T}=\mathcal{N}\cup\{\mathsf{root}\}$. *Then, the iterates* $\left(\boldsymbol{\mu}^{(s)}: s\geq 1\right)$ *generated by Step 1 of the* A-MM *algorithm satisfy,*

$$\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s)}\right) - \mathsf{NegLog}\left(\boldsymbol{\mu}^{*}\right) \leq \frac{2D^2}{s},$$

where $\boldsymbol{\mu}^*$ *denotes the unique optimal solution to the* MLE problem, *and* $D\geq 0$ *defined as*

$$D = \max\left\{\|\boldsymbol{\mu}-\boldsymbol{\mu}^*\|_\infty : \boldsymbol{\mu}\in\mathsf{Dom}_1,\, \mathsf{NegLog}(\boldsymbol{\mu})\leq\mathsf{NegLog}(\boldsymbol{\mu}^{(0)})\right\}$$

is finite. Starting from the same initial solution, the GD iterates $\left(\boldsymbol{\mu}_{\mathsf{GD}}^{(s)}\colon s \geq 1\right)$ *with step size* $1/L$*—where* L *is the smoothness constant of* $\mathsf{NegLog}(\boldsymbol{\mu})$*—satisfy,*

$$\mathsf{NegLog}\left(\boldsymbol{\mu}_{\mathsf{GD}}^{(s)}\right) - \mathsf{NegLog}\left(\boldsymbol{\mu}^*\right) \leq \frac{2nL \cdot D^2}{s}.$$

Recall that the optimal solution $\boldsymbol{\mu}^*$ exists and is unique because of the identification condition in Theorem 3.2. The proof of the theorem is presented in Appendix E.5. Our result shows that the MM algorithm can be factor n faster than the standard GD algorithm (recall that as shown in Theorem 4.5, the smoothness constant $L \leq 1/\lambda_{\mathsf{lower}}^2 = 1$ for the MNL model, which is tight up to constant factors). Our result is the first to establish a stronger convergence rate guarantee for the MM algorithm over the standard GD algorithm. Previously, the best known result in the literature was from Vojnovic et al. (2020), who established that both MM and GD algorithms have the same linear convergence rates (up to constant factors), but only for the special case of the Bradley-Terry model (MNL with pairwise comparisons) and with the additional assumption that the negative log-likelihood function is strongly convex. In the absence of this additional assumption, their result reduces to establishing that the MM and GD algorithms have the same sub-linear rates (modulo constant factors). Apart from this result, there isn't substantial work on analyzing the convergence properties of the MM algorithm. Hunter (2004) and, Abdallah and Vulcano (2021) both showed that the algorithm converges to the unique maximum likelihood estimate as long as the strongly connected condition in Theorem 3.2 is satisfied, but no rate was given. Most existing work, however, observed that the MM algorithm converged faster than gradient descent and off-the-shelf optimization routines in practice. Our result now provides theoretical support for this empirical observation, especially in large-scale settings.

We now describe our proposed A-MM method in more detail.

4.3 MM updates for generating improving solutions

Our algorithm is based on the popular "optimization transfer" meta-heuristic (Lange et al. 2000). The basic idea is to reduce the minimization of a (difficult) objective function into minimizing a sequence of (simpler) surrogate functions, such that each iteration of the algorithm provides an improving solution. The surrogate function is chosen to *majorize*, that is, upper bound the original function around the solution in each iteration; a precise definition of a majorizing surrogate is given in Definition 4.7. The surrogate function is then minimized either exactly or approximately to obtain a new solution, and the process is repeated until a stopping condition is met. Hunter

and Lange (2000) coined the term "majorization-minimization" (MM) to refer to such iterative optimization algorithms.[7]

The art of designing a good MM algorithm lies in the design and optimization of the surrogate function, which allows one to trade off the total number of iterations of the algorithm with the difficulty of optimizing the surrogate function in each iteration. A simple-to-optimize surrogate function may serve as a poor approximation of the original function and thereby require many iterations to converge. On the other hand, although a complex surrogate function may be hard to optimize, it may result in fewer total iterations. How to strike the optimal trade-off is a priori unclear, but Hunter and Lange (2004) proposed several techniques that exploit the convexity of the original function to design practically good surrogate functions.

With the above background, we now formally define a surrogate majorizing function. Consider the following optimization problem:

$$\min_{\boldsymbol{x}\in\mathcal{X}} f(\boldsymbol{x}),$$

where $\mathcal{X}$ is a convex subset in Euclidean space. The MM approach relies on constructing a majorizing surrogate for f, defined as

Definition 4.7 (Majorizing Surrogate) *A function $g(\cdot\,|\bar{\boldsymbol{x}})$ is a majorizing surrogate for f at $\bar{\boldsymbol{x}}\in\mathcal{X}$ if $g(\cdot\,|\,\bar{\boldsymbol{x}})$ is continuously differentiable, strictly convex, and*

$$f(\boldsymbol{x}) \;\le\; g(\boldsymbol{x}\,|\bar{\boldsymbol{x}}) \;\forall\; \boldsymbol{x}\in\mathcal{X} \quad \textit{and} \quad f(\bar{\boldsymbol{x}}) \;=\; g(\bar{\boldsymbol{x}}\,|\bar{\boldsymbol{x}})\;.$$

Starting from an initial estimate $\boldsymbol{x}^{(0)}$, the MM algorithm performs the following update in each iteration $s\ge 0$:

$$\boldsymbol{x}^{(s+1)} \leftarrow \operatorname*{arg\,min}_{\boldsymbol{x}\in\mathcal{X}} \; g(\boldsymbol{x}\,|\boldsymbol{x}^{(s)}) \qquad \text{(MM update)}$$

Note that since the majorizing surrogate $g(\cdot\,|\boldsymbol{x}^{(s)})$ is strictly convex, the arg min above is well-defined and unique. The majorizing property of the surrogate function guarantees that the MM update can only improve the objective value:

$$f(\boldsymbol{x}^{(s+1)}) \le g(\boldsymbol{x}^{(s+1)}\,|\boldsymbol{x}^{(s)}) \le g(\boldsymbol{x}^{(s)}\,|\boldsymbol{x}^{(s)}) = f(\boldsymbol{x}^{(s)}). \tag{4}$$

We leverage the MM principle to solve the μ update-subproblem and δ update-subproblem. To reduce the computational burden, we only perform one MM update for each of the subproblems, which is sufficient to guarantee an improving solution in each iteration of the A-MM algorithm.

[7] MM also refers to minorization-maximization, when the original problem requires maximization instead of minimization. Unlike the majorization step, the minorization step involves lower bounding the original objective function.

4.3.1 An improving solution for the μ update-subproblem. We begin by constructing a majorizing surrogate for the mapping $\boldsymbol{\mu} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$ for any fixed $\boldsymbol{\delta} \in \mathsf{Dom}_2$. Given any $\boldsymbol{\mu} \in \mathsf{Dom}_1$ and $\boldsymbol{\delta} \in \mathsf{Dom}_2$, for each $\ell \in \mathcal{N}$, define

$$a_\ell(\boldsymbol{\mu}, \boldsymbol{\delta}) \;=\; \exp\left(-\delta_{[\mathsf{pa}(\ell)]}\right) \cdot \partial \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})/\partial \mu_\ell + \mathsf{sales}_\ell, \tag{5}$$

where recall that sales_ℓ denotes the fraction of sales of product ℓ. The following theorem exhibits a majorizing surrogate by exploiting the strict convexity of the mapping $\boldsymbol{\mu} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$, which was established in Theorem 4.1:

Theorem 4.8 (Exploiting Strict Convexity to Construct a Majorizing Surrogate) *For any $\bar{\boldsymbol{\delta}} \in \mathsf{Dom}_2$, a majorizing surrogate for the function $\mathsf{NegLog}(\boldsymbol{\mu}, \bar{\boldsymbol{\delta}})$ at $\bar{\boldsymbol{\mu}} \in \mathsf{Dom}_1$ is given by the separable function $G(\boldsymbol{\mu} \,|\, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) = \sum_{\ell \in \mathcal{N}} G_\ell\left(\mu_\ell \,|\, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}\right)$, where for each $\ell \in \mathcal{N}$, the function $G_\ell\left(\cdot \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}}\right)$ is strictly convex and*

$$\arg\min_{x \in \mathbb{R}}\ G_\ell\left(x \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}}\right) \;=\; \bar{\mu}_\ell \;+\; \exp\left(-\bar{\delta}_{[\mathsf{pa}(\ell)]}\right) \cdot \log\left(\mathsf{sales}_\ell \,/\, a_\ell\left(\bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}}\right)\right).$$

The proof in Appendix E.6 gives an explicit expression for each $G_\ell\left(\cdot \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}}\right)$. It also shows that $a_\ell(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) > 0$ for all $\bar{\boldsymbol{\mu}} \in \mathsf{Dom}_1$ and $\bar{\boldsymbol{\delta}} \in \mathsf{Dom}_2$ so that the minimizer aboue is well-defined. We constructed the above majorizing surrogate because (a) its (unique) minimizer can be computed quickly in closed form, and (b) the update step reduces to the standard MM update for estimating the classical multinomial logit (MNL) model, originally proposed by Hunter (2004) and extended recently to account for censored demand by Abdallah and Vulcano (2021). Specifically, under the MNL model, $\mathsf{T} = \mathcal{N} \cup \{\mathsf{root}\}$. Then, by plugging in the expression for $\partial \mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}})/\partial \mu_\ell$ from Lemma C.2 in Appendix C and using the fact that $\bar{\delta}_{[\mathsf{pa}(\ell)]} = \bar{\delta}_{[\mathsf{root}]} = 0$, it follows that for all $\ell \in \mathcal{N}$,

$$a_\ell(\bar{\boldsymbol{\mu}}) = \exp(-\bar{\delta}_{[\mathsf{root}]}) \cdot \left(\frac{1}{Q}\sum_{q=1}^{Q}\ \mathbb{1}_{\{c^q \in \mathsf{T}_{\mathsf{root}}\}} \psi^q_{\mathsf{root} \to \ell}(\bar{\boldsymbol{\mu}}) - \mathsf{sales}_\ell\right) + \mathsf{sales}_\ell \;=\; \frac{1}{Q}\sum_{q=1}^{Q} \psi^q_{\mathsf{root} \to \ell}(\bar{\boldsymbol{\mu}})$$

where we drop the reference to $\boldsymbol{\delta}$ since there are no non-leaf nodes in T. Since $\psi^q_{\mathsf{root} \to \ell}(\bar{\boldsymbol{\mu}})$ is the probability of purchasing product ℓ from offer-set S^q, $a_\ell(\bar{\boldsymbol{\mu}})$ represents the *estimated* fraction of sales of product ℓ based on the parameter $\bar{\boldsymbol{\mu}}$. Then, using the expression of the minimizer from Theorem 4.8, it follows that

$$\arg\min_{x \in \mathbb{R}} G_\ell\left(x \,|\, \bar{\boldsymbol{\mu}}\right) \;=\; \bar{\mu}_\ell \;+\; \log \frac{\mathsf{sales}_\ell}{\frac{1}{Q}\sum_{q=1}^{Q} \psi^q_{\mathsf{root} \to \ell}(\bar{\boldsymbol{\mu}})} \;=\; \bar{\mu}_\ell \;+\; \log \frac{\text{Actual sales of product } \ell}{\text{Estimated sales of product } \ell \text{ under } \bar{\boldsymbol{\mu}}},$$

which is consistent with the standard MM update for the MNL model.

Now, the MM update in the context of the μ update-subproblem takes the form $\mu_\ell^{(s+1)} = \arg\min_{x \in \mathbb{R}} G_\ell\left(x \,|\, \boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}\right)$ for all $\ell \in \mathcal{N}$. However, this minimizer may not be feasible,

i.e. belong to Dom_1. To ensure that the next iterate remains feasible, we subtract the mean utility of product 1 from all the utilities, which results in the update presented in Step 1 of the A-MM algorithm. This transformation does not affect the negative log-likelihood value since it is shift-invariant in the mean utilities $\boldsymbol{\mu}$, and therefore the descent property (4) is still satisfied.

The MM update also has a very intuitive interpretation. Suppose that $\boldsymbol{\mu}^{(s)}$ is a stationary point of the problem $\min_{\boldsymbol{\mu}\in\mathsf{Dom}_1} \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}^{(s)})$. Then, from Definition 4.2, it follows that $\partial\mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})/\partial\mu_\ell = 0$ for all $\ell \in \mathcal{N}\setminus\{1\}$. Since $\sum_{\ell\in\mathcal{N}} \partial\mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})/\partial\mu_\ell = 0$ (see proof of Lemma E.3 in Appendix E.3), this also implies that $\partial\mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})/\partial\mu_1 = 0$. Then, it is easy to check that $\mu_\ell^{(s+1)} = \mu_\ell^{(s)}$ for all $\ell \in \mathcal{N}$, or in other words, the MM update leaves the iterate unchanged. On the other hand, suppose there exists some $\ell \in \mathcal{N}\setminus\{1\}$ such that $\partial\mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})/\partial\mu_\ell \neq 0$. If $\partial\mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})/\partial\mu_\ell > 0$, then $a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) > \mathsf{sales}_\ell$ and the MM update decreases the estimate of μ_ℓ; otherwise the MM update increases the estimate of μ_ℓ. Therefore, the MM update iteratively adjusts the value of μ_ℓ, always in the right direction, until the value converges to a stationary point.

4.3.2 An improving solution for the δ update-subproblem. Similar to the update for the mean utility parameters above, we design an MM method for solving the δ update-subproblem by leveraging the DSC representation from Theorem 4.1. We begin by exhibiting a majorizing surrogate for the mapping $\boldsymbol{\delta} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$ for any fixed $\boldsymbol{\mu} \in \mathsf{Dom}_1$:

Lemma 4.9 (Exploiting DSC Representation to Construct a Majorizing Surrogate) *For any $\bar{\boldsymbol{\mu}} \in \mathsf{Dom}_1$, a majorizing surrogate for the function $\mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta})$ at $\bar{\boldsymbol{\delta}} \in \mathsf{Dom}_2$ is given by the following strictly convex function:*

$$H(\boldsymbol{\delta} \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}}) = F_1(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta}) - F_2(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) - \sum_{j\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})} \frac{\partial F_2}{\partial \delta_j}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) \cdot \left(\delta_j - \bar{\delta}_j\right)$$

where F_1, F_2 are as defined in Theorem 4.1.

Proof. Recall from Theorem 4.1 that $\mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta}) = F_1(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta}) - F_2(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta})$ for all $\boldsymbol{\delta} \in \mathsf{Dom}_2$, where both $F_1(\bar{\boldsymbol{\mu}}, \cdot)$ and $F_2(\bar{\boldsymbol{\mu}}, \cdot)$ are strictly convex on Dom_2. By applying the sub-gradient inequality to the strictly convex function $F_2(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta})$ at $\boldsymbol{\delta} = \bar{\boldsymbol{\delta}}$, it follows that

$$\mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta}) \leq F_1(\bar{\boldsymbol{\mu}}, \boldsymbol{\delta}) - F_2(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) - \sum_{j\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})} \frac{\partial F_2}{\partial \delta_j}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) \cdot \left(\delta_j - \bar{\delta}_j\right) \ \forall\ \boldsymbol{\delta} \in \mathsf{Dom}_2$$
$$\text{with equality if and only if} \quad \boldsymbol{\delta} = \bar{\boldsymbol{\delta}}\ . \quad (6)$$

The result then follows from the definition of a majorizing surrogate. ■

Unlike the majorizing surrogate $G(\cdot \,|\, \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}})$ of Section 4.3.1, the function $H(\cdot \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}})$ does not admit a closed form optimizer. However, since we only need to generate an improving solution to ensure the descent property (4), it is enough to obtain an approximate minimizer for the problem $\min_{\boldsymbol{\delta} \in \mathsf{Dom}_2} H(\boldsymbol{\delta} \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}})$. In particular, we can obtain an improving solution by performing one iteration of projected gradient descent (PGD), with the step size chosen by linesearch. Given any $\boldsymbol{x} \in \mathbb{R}^{|\mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})|}$, it is easy to see that the projection onto Dom_2 can be computed efficiently as $((x_j)^+ :\ j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\}))$, where recall that $(x_j)^+ = \max(x_j, 0)$. This combined with the fact that $\nabla H(\boldsymbol{\delta}^{(s)} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)}) = \nabla_{\boldsymbol{\delta}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)})$, gives the update outlined in Step 2 of the A-MM algorithm. We emphasize that the change of variables from the $\boldsymbol{\lambda}$ to the $\boldsymbol{\delta}$ domain is crucial as it ensures that the projection step can be solved in closed form, unlike projection onto the original domain $\mathcal{D}_2$.

It is worth noting that Step 2 of the A-MM algorithm is identical to that obtained when performing a PGD update for $\boldsymbol{\delta}$ directly on the negative log-likelihood objective, but with (possibly) different step sizes. This is precisely why the improvement bounds for both approaches are the same (ignoring constants) as shown in Lemma F.1 in Appendix F. However, similar to the case for the $\boldsymbol{\mu}$ update earlier, the improvement bound for PGD relies on tuning the step size, which can be challenging in practice since the negative log-likelihood is non-convex in $\boldsymbol{\delta}$. The A-MM update, on the other hand, always guarantees the improvement bound since the optimal step size can be computed in closed form, as established in Theorem 4.3 earlier.

5. Numerical Results

In this section, we conduct two numerical studies to evaluate the performance of our proposed A-MM estimation algorithm, one on synthetic data and the other on real data. In both studies, we compare our method against two standard benchmarks for estimating the parameters of a tree logit model: the standard projected gradient descent method as well as the Artelys Knitro[8] solver, a state-of-the-art commercial package for solving large-scale nonlinear mathematical optimization problems. We refer to the two benchmarks as PGD and `Knitro`, respectively, in the remainder of the section.

5.1 Synthetic Data

We first conduct a simulation study to test the performance of the different methods under a range of problem sizes. We find that our method obtains significantly lower negative log-likelihood values than the two benchmark methods, especially on harder problem instances—those with larger

[8] https://www.artelys.com/docs/knitro

numbers of nodes and smaller values for the nest dissimilarity parameters. Moreover, the performance of our method remains stable across the problem instances, converging to solutions that are close to optimal even for larger problem sizes. In contrast, given similar computational budget, the benchmarks struggle to get close to the optimal solution even for moderately-sized problems.

5.1.1 Setup We carry out the simulations in four steps: we (a) sample ground-truth tree logit instances, (b) sample choice data from these instances, (c) fit tree logit models to the sampled choice data using the A-MM, `Knitro` and PGD methods, and (d) compare the three methods on the (average) negative log-likelihood values obtained under comparable average running times. In particular, we test how well these methods scale to large problem sizes (measured in terms of the numbers of non-leaf and leaf nodes in the tree) and to difficult instances previously identified in the literature (Koppelman and Bhat 2006, Daganzo and Kusnic 1990), in which the nest dissimilarity parameters become close to zero. For that, we generate different ground-truth instances by varying the out-degree (or the number of children) r of each non-leaf node, the height H of the tree, and the lower bound $\lambda_{\mathsf{lower}} \in (0,1)$ on the nest dissimilarity parameters. We generate 24 instances by varying the tuple $(r, H, \lambda_{\mathsf{lower}})$ over the set $\{5,6,7,8\} \times \{4,5\} \times \{0.01, 0.10, 0.50\}$.

For each tuple $(r, H, \lambda_{\mathsf{lower}})$, we consider the tree logit model corresponding to the perfect r-ary tree of height H, where each non-leaf node has r children, and generate 100 instances randomly as follows:

1. For each leaf node $\ell \neq 1$, we sample the mean utility value μ_ℓ independently and uniformly at random (u.a.r.) from the interval $[0,1]$. We set $\mu_1 = 0$.
2. For the root node, we set $\lambda_{\mathsf{root}} = 1$. Then, starting from the children of the root node, we recursively sample the nest dissimilarity parameter λ_j independently and u.a.r. from the interval $[\lambda_{\mathsf{lower}}, \lambda_{\mathsf{pa}(j)}]$, for each non-leaf node j. This recursive sampling procedure ensures that the sampled nest dissimilarity parameters always respect the feasibility constraints $0 < \lambda_{\mathsf{lower}} \leq \lambda_j \leq \lambda_{\mathsf{pa}(j)} \leq 1$, which ensures that the associated model satisfies the RUM principle.
3. Once the parameters of the model are sampled, we generate choice data for 60 offer sets. We randomly generate an offer set by including each product with probability 90%, independently of all the other products. For every sampled offer-set $\mathcal{S}$, we simulate the choices of 100 customers, where for each customer, we sample her choice $\ell \in \mathcal{S}$ independently with probability $\mathbb{P}_\ell(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$.

5.1.2 Implementation Details and Performance Measures. For each ground-truth instance $m = 1, 2, \ldots, 100$, we fit tree logit models to the sampled choice data using the A-MM, PGD and `Knitro` methods. We use the MATLAB implementation of the `Knitro` solver. Knitro implements four state-of-the-art interior-point and active-set methods for solving continuous, nonlinear optimization problems. We run all the four Knitro algorithms in parallel, and choose the one that

achieves the best solution. For the PGD benchmark, we leverage Knitro to solve the projection step (3), and projected backtracking linesearch (Nocedal and Wright 2006b) to compute the step size in each iteration. While the linesearch in Step 2 of our proposed A-MM algorithm can be solved in closed form (as shown in Theorem 4.3), to reduce the computational burden, we also leverage the projected backtracking linesearch algorithm to compute an approximate step size. All methods are initialized at the same starting point, $\boldsymbol{\mu}^{(0)} = \mathbf{0}$ and $\boldsymbol{\lambda}^{(0)} = \mathbf{1}$.

We carry out our numerical simulations on the NYU High Performance Computing (HPC) cluster. Ground-truth instances are generated in parallel on 100 cluster nodes, each with 2 Intel Xeon Gold 6148 2.4GHz CPUs and 187 GB of memory. To ensure that the three methods run in comparable CPU times, we run A-MM for 400 iterations, `Knitro` for 8 iterations, and PGD for 50 iterations. By doing so, the A-MM method (with avg. 1.5-hour runtime per instance) is more than 3× faster than the `Knitro` method (with avg. 5-hour runtime per instance) for larger problem sizes with more than 5000 products. The runtime of the A-MM method is impressive, especially considering the fact that the implementation of `Knitro` has been optimized in MATLAB, whereas our A-MM method is implemented in pure MATLAB, without any under-the-hood optimizations. Indeed, a more optimized implementation of our method can scale to even larger problem sizes.

While our proposed A-MM method always outputs feasible solutions, we observed that both `Knitro` and PGD often yield infeasible estimates due to rounding errors. In particular, the constraint $\lambda_j \le \lambda_{\mathsf{pa}(j)}$ is violated for some non-leaf node j so that the final model lies outside the RUM class. Aside from rounding issues, we also find that the `Knitro` method struggles to converge to a feasible solution on larger problem instances, even with all four of its state-of-the-art algorithms running in parallel.

We measure the performance of each method by the gap between the negative log-likelihood under the true model parameters and the negative log-likelihood value obtained by the method. For any ground-truth instance $(r, H, \lambda_{\mathsf{lower}})$, let $\mathsf{NegLog}^{\mathsf{PGD},m}$, $\mathsf{NegLog}^{\texttt{Knitro},m}$ and $\mathsf{NegLog}^{\mathsf{A\text{-}MM},m}$ denote the negative log-likelihood objective in the MLE problem evaluated at the parameters obtained by the PGD, `Knitro`, and A-MM methods, respectively for each problem instance $m = 1, 2, \ldots, 100$. Further, let $\mathsf{NegLog}^{\mathrm{true},m}$ denote the negative log-likelihood of the sampled choice data for instance m under the ground-truth model parameters. Then, we measure the performance of each method by the average NegLogGap, defined as:

$$\mathsf{NegLogGap}^{\mathsf{PGD}} = \frac{1}{100}\sum_{m=1}^{100}\left(\mathsf{NegLog}^{\mathsf{PGD},m} - \mathsf{NegLog}^{\mathrm{true},m}\right),$$

$$\mathsf{NegLogGap}^{\texttt{Knitro}} = \frac{1}{100}\sum_{m=1}^{100}\left(\mathsf{NegLog}^{\texttt{Knitro},m} - \mathsf{NegLog}^{\mathrm{true},m}\right) \quad \text{and}$$

$$\mathsf{NegLogGap}^{\mathsf{A\text{-}MM}} = \frac{1}{100}\sum_{m=1}^{100}\left(\mathsf{NegLog}^{\mathsf{A\text{-}MM},m} - \mathsf{NegLog}^{\mathrm{true},m}\right),$$

where smaller values for the gaps are preferred. In Appendix I, we also compare the performance of each method based on the root mean squared error (RMSE) between the estimated and the ground-truth choice probabilities.

5.1.3 Results and Discussion We report the results in a table shown in Figure 3. The first, second and fifth columns report the out-degree of each non-leaf node, height of the tree, and the lower bound on the nest dissimilarity parameters, respectively. The third and fourth columns report the number of products and the number of nodes in the trees, respectively. Columns six to eight report the average NegLogGap values obtained for the A-MM, PGD and Knitro methods, respectively. Columns nine and ten report the improvements $\mathsf{NegLogGap}^{\mathsf{PGD}} - \mathsf{NegLogGap}^{\mathsf{A\text{-}MM}}$ and $\mathsf{NegLogGap}^{\mathtt{Knitro}} - \mathsf{NegLogGap}^{\mathsf{A\text{-}MM}}$, respectively in the negative log-likelihood value the A-MM method obtains over the two benchmarks. Finally, the last two columns report the percentage of instances in which the A-MM method obtains a lower negative log-likelihood value than the PGD and Knitro methods, respectively.

Comparing the A-MM method with the benchmarks, we draw the following conclusions:

1. *Our method obtains significantly better negative log-likelihood values.* The A-MM method obtains lower average negative log-likelihood values than the PGD and Knitro methods for all problem sizes, indicated by the positive numbers under the "NegLog Impr." column. Moreover, the improvements are larger for larger problem sizes, with differences in the negative log-likelihood values frequently over 50 (on a logarithmic scale). In fact, for larger problem sizes, our method outperforms the benchmarks for all 100 instances, indicated by the value 100 under the "% better" column.
2. *Our performance is robust to problem sizes.* It can be seen that the NegLogGap value of the A-MM method remains stable across a range of problem sizes, indicating that our method can converge to a solution that is close to optimal even for larger problem sizes. By contrast, both the PGD and Knitro method struggle to get close to the optimal solution for larger problem sizes, with Knitro unable to complete even a single iteration.
3. *Nest dissimilarity parameters closer to zero result in harder instances.* For a given problem size (i.e. height of the tree and the degree of each non-leaf node), we note that the performance of all the methods suffer as λ_{lower} decreases from 0.5 to 0.01. This trend indicates that smaller values of nest dissimilarity parameters are harder to estimate from choice data, which is consistent with findings in existing literature (Koppelman and Bhat 2006, Daganzo and Kusnic 1990). However, the NegLogGap values for the A-MM method degrade more gradually, leading to stark improvements (over 100) in the negative log-likelihood values over the benchmarks for smaller values of λ_{lower}.

Degree	Height	# Prods.	# Nodes	λ_{lower}	NegLogGap			NegLog Impr.		% better	
					A-MM	PGD	Knitro	over PGD	over Knitro	over PGD	over Knitro
5	4	625	781	0.50	2.6	5.2	20.3	2.6	17.7	76	90
				0.10	7.8	8.0	74.2	0.2	66.4	45	85
				0.01	9.9	11.7	90.7	1.8	80.8	55	87
	5	3,125	3,906	0.50	3.6	16.2	35.9	12.6	32.3	100	100
				0.10	13.2	24.4	51.7	11.2	38.5	96	100
				0.01	21.6	35.3	65.4	13.7	43.8	88	94
6	4	1,296	1,555	0.50	2.5	15.4	16.4	12.9	13.9	100	100
				0.10	8.5	20.0	68.2	11.5	59.7	93	100
				0.01	11.1	31.6	81.6	20.5	70.5	100	100
	5	7,776	9,331	0.50	3.3	40.2	215.4	36.9	212.1	100	100
				0.10	12.8	56.0	499.6	43.2	486.8	100	100
				0.01	20.9	86.7	515.7	65.8	494.8	100	100
7	4	2,401	2,801	0.50	2.4	31.2	30.3	28.8	27.9	100	100
				0.10	8.6	49.3	90.1	40.7	81.5	100	100
				0.01	11.4	59.5	95.8	48.1	84.4	100	100
	5	16,807	19,608	0.50	2.9	71.5	—	68.6	—	100	—
				0.10	12.1	91.1	—	79.0	—	100	—
				0.01	19.0	138.8	—	119.8	—	100	—
8	4	4,096	4,681	0.50	2.2	60.6	—	58.4	—	100	—
				0.10	8.0	74.5	—	66.5	—	100	—
				0.01	10.5	106.2	—	95.7	—	100	—
	5	32,768	37,449	0.50	2.5	115.7	—	113.2	—	100	—
				0.10	10.7	128.6	—	117.9	—	100	—
				0.01	17.2	210.1	—	192.9	—	100	—

Figure 3 Comparison of the performances of the PGD, Knitro and our proposed A-MM method in fitting tree logit models to choice data. The columns "Degree", "Height", and λ_{lower} report the degree of each non-leaf node, the height of the tree, and the lower bound on the nest dissimilarity parameters, respectively. The columns "# Prods." and "# Nodes" report the number of products and the number of nodes in the tree, respectively. The columns under A-MM, PGD and Knitro report the average NegLogGap for each method. Recall that smaller values for the gaps are preferred. The columns under "NegLog Impr." report the average improvement in the negative log-likelihood value that our A-MM method achieves over the PGD and Knitro benchmarks. Finally, the columns under "% better" report the percentage of instances in which the A-MM method obtains a lower NegLog value than each benchmark. The Knitro benchmark is unable to complete even a single iteration for large problem sizes, and we use "—" to denote such instances. All the "NegLog Impr." numbers are significantly different from zero at the 1% significance level under a paired samples t-test.

In summary, we observe that our proposed A-MM method easily scales to tree structures with thousands of nodes, unlike the PGD and `Knitro` benchmarks, which struggle to find good quality solutions even for moderately sized problems. The improvement in performance is quite dramatic for harder problem instances—those with larger numbers of nodes and smaller values of nest dissimilarity parameters.

5.2 Real Data: SUSHI Preference Dataset

In this subsection, we evaluate the performance of the different methods on a real-world dataset containing preference orderings over different sushi varieties (Kamishima 2003). The preference data was collected using a survey, where 5000 individuals were asked to rank their top 10 sushi varieties from a collection of 100 sushi types. Each sushi variety is associated with descriptive features such as its category (maki or non-maki), oiliness level, price, etc. We choose this dataset since the rank-order information enables the simulation of customer choices on arbitrary offer-sets, and the feature information provides a way to define a natural tree structure. Our results show that the A-MM method is robust to initialization and achieves significantly lower negative log-likelihood values on both training and test data, compared to the two benchmarks.

5.2.1 Setup and Performance Measure. We randomly generate 400 problem instances as follows. We first sample 1000 offer-sets, where each offer-set is generated randomly by including each sushi type with probability 70%, independently of all the other types. Once the offer-sets are sampled, we offer each of them to the 5000 individuals in our dataset. Each individual chooses the available sushi type that is ranked highest in her top-10 ordering—if none of the products in the ranking is part of the offer set, then we suppose that the individual chooses the no-purchase option. As a result, we simulate 5000 transactions for each offer-set, for a total of $1000 \times 5000 = 5$M transactions for each problem instance. Next, we divide the 1000 offer sets random into a training (700) and test (300) group, and fit a tree logit model to the choice data from the training group using the PGD, `Knitro` and A-MM methods. The tree structure is shown in Appendix H and is defined based on the available features for each sushi variety. To evaluate the benefit of estimating a tree logit model for this dataset, we also fit the standard multinomial logit (MNL) model using each method.

To gauge the robustness of the different methods to the initial solution, we fit tree logit models from two starting points: (a) *0/1 start*, in which we set $\boldsymbol{\mu}^{(0)} = \mathbf{0}$, $\boldsymbol{\lambda}^{(0)} = \mathbf{1}$, and (b) *warm start*, in which we set $\boldsymbol{\mu}^{(0)} = \boldsymbol{\mu}^{(\text{MNL})}, \boldsymbol{\lambda}^{(0)} = \mathbf{1}$, where $\boldsymbol{\mu}^{(\text{MNL})}$ are the mean utilities estimated when fitting the MNL model under the corresponding method. To keep the running times of the different methods comparable, we run the A-MM and PGD methods for 400 iterations, and `Knitro` for 50 iterations. Under this setting, the A-MM method (155 sec avg. runtime per instance) is still faster

than the Knitro (160 sec avg. runtime per instance) and PGD (213 sec avg. runtime per instance) benchmarks.

Since we do not have access to the "true" model parameters in this case, we measure the performance of each method by the improvement obtained over fitting an MNL model. In particular, let $\mathsf{NegLog}_{\text{Train}}^{\text{algo},m}$ and $\mathsf{NegLog}_{\text{Test}}^{\text{algo},m}$ denote respectively, the negative log-likelihood value on the training and test offer-sets for problem instance m, evaluated under the parameters obtained by fitting a tree logit model using method algo $\in \{\text{A-MM}, \text{PGD}, \texttt{Knitro}\}$. Similarly, let $\mathsf{NegLog}_{\text{Train}}^{\text{MNL},m}$ and $\mathsf{NegLog}_{\text{Test}}^{\text{MNL},m}$ denote respectively, the negative log-likelihood value on the training and test offer-sets for problem instance m, obtained when fitting the MNL model.[9] Then, we report the average improvement in the negative log-likelihood values on the training and test offer-sets separately, computed as the following:

$$\mathsf{NegLogImp}_{\text{Train}}^{\text{algo}} = \frac{1}{400}\sum_{m=1}^{400}\left(\mathsf{NegLog}_{\text{Train}}^{\text{MNL},m} - \mathsf{NegLog}_{\text{Train}}^{\text{algo},m}\right)$$

$$\mathsf{NegLogImp}_{\text{Test}}^{\text{algo}} = \frac{1}{400}\sum_{m=1}^{400}\left(\mathsf{NegLog}_{\text{Test}}^{\text{MNL},m} - \mathsf{NegLog}_{\text{Test}}^{\text{algo},m}\right).$$

Note that larger values of NegLogImp are preferred. In Appendix I, we also compare the performance of each method based on the root mean squared error (RMSE) between the estimated and observed choice fractions.

Initialization	Train NegLogImp			Test NegLogImp			Impr. over PGD		Impr. over Knitro	
	A-MM	Knitro	PGD	A-MM	Knitro	PGD	Train	Test	Train	Test
0/1 start	628.7	385.0	-2,396.0	266.9	161.7	-1,034.6	3,024.7	1,301.5	243.7	105.2
warm start	628.7	566.2	379.5	266.9	240.3	161.9	249.2	105.1	62.5	26.6

Figure 4 Comparison of the performances of PGD, Knitro and our proposed A-MM method in fitting tree logit models to the Sushi Preference Dataset. The first (resp. second) row reports the performance when the methods are initialized using *0/1 start* (resp. *warm start*). The second, third and fourth columns report the NegLogImp value of A-MM, Knitro and PGD on the training data, while the fifth, sixth and seventh columns report the corresponding NegLogImp values on the test data. Recall that larger values for the gaps are preferred. The eight and tenth columns report the average improvement in the negative log-likelihood value of the A-MM method over the PGD and Knitro benchmarks, respectively, on the training data. The corresponding improvements on the test data are reported in columns nine and eleven. All numbers under the "Impr." columns are significantly different from zero at the 1% significance level under a paired sample t-test.

[9] We observed that both the A-MM and Knitro methods converged to the optimal solution for each problem instance, and therefore, use the negative log-likelihood under either of the two methods as the baseline.

5.2.2 Results and Discussion We report the results in a table shown in Figure 4. The first row shows the performance under *0/1 start*, and the second under *warm start*. The second, third and fourth (resp. fifth, sixth and seventh) columns report the average NegLogImp value of the A-MM, `Knitro` and PGD methods on the training (resp. test) data. The eighth and tenth columns report the improvements $\text{NegLogImp}^{\text{A-MM}}_{\text{Train}} - \text{NegLogImp}^{\text{PGD}}_{\text{Train}}$ and $\text{NegLogImp}^{\text{A-MM}}_{\text{Train}} - \text{NegLogImp}^{\texttt{Knitro}}_{\text{Train}}$ respectively, in the negative log-likelihood value that the A-MM method obtains over the PGD and `Knitro` benchmarks on the training data. Similarly, the ninth and eleventh columns report the improvements on the test data. Comparing our method to the two bechmarks, we draw the following conclusions:

1. *Tree logit model performs better than fitting the MNL model.* We find that fitting a tree logit model results in significantly lower negative log-likelihood on both the training and test data, indicated by the positive NegLogImp values.[10] In particular, fitting a tree logit model with the A-MM method achieves an average improvement of 628.7 (resp. 266.9) in the negative log-likelihood over the MNL model on the training (resp. test) data—note that these values are on the logarithmic scale. This finding suggests that for this dataset, capturing correlations in the utilities of different sushi varieties helps to better explain the customer choice behavior.
2. *Our method outperforms the benchmarks on both training and test data.* The A-MM method achieves lower negative log-likelihood values compared to the PGD and `Knitro` benchmarks, on both the training and test data. In particular, even under the *warm start* initialization, the A-MM method achieves an improvement of 105.1 and 26.6 in the negative log-likelihood values over the PGD and `Knitro` benchmarks respectively, on the test data. The improvements are larger under *0/1 start*. The improved prediction accuracy highlights the importance of using the right estimation method in practice, since demand predictions from the choice model feed into downstream decision problems like assortment and pricing optimization.
3. *Our performance is robust to the initialization.* It can be seen that the NegLogImp values achieved by the A-MM method are the same under both *0/1 start* and *warm start*, indicating that our method is robust to the initial solution. In contrast, both `Knitro` and PGD are sensitive to the initial solution. In fact, the PGD benchmark with *0/1 start* achieves a higher negative log-likelihood value than the MNL model, indicating that it is unable to converge to a good solution under the given computation budget.

6. Conclusion and Future Research

Tree logits have been used to model customer choice in many applications. They also result in tractable operational decision problems. Given their significance, this paper considers the problem

[10] For the PGD benchmark, the NegLogImp values are negative under *0/1 start* and we discuss the reason below.

of estimating the parameters of a tree logit model from choice data. We propose a novel variable transformation and exploit the resulting structure in the negative log-likelihood objective to design a scalable estimation algorithm based on the majorization-minimization (MM) framework. We analyze the convergence rate of our estimation algorithm and show that it can converge faster than gradient descent on certain problem instances. Using both synthetic and real data, we show that our proposed method achieves significantly lower negative log-likelihood values compared to gradient descent as well as the state-of-the-art Knitro solver, given comparable running times.

Our study opens up exciting new frontiers. On the implementation front, we can potentially speed up our algorithm by exploring its connections to the back propagation method. A key computational step in our algorithm is the computation of the gradients $(\partial \mathsf{NegLog}/\partial \mu_\ell : \ell \in \mathcal{N})$ and $(\partial \mathsf{NegLog}/\partial \delta_j : j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\}))$. These gradient computations may be performed more efficiently using the back propagation method on the tree. If such a connection can be established, we can take advantage of the high-performance features offered by existing computational frameworks, thus enabling our algorithm to scale to millions of nodes.

On the algorithmic front, designing stochastic variants of our method, that compute the gradient of the parameters on a single transaction as opposed to all the transactions (e.g., see, Johnson and Zhang (2013) and Defazio et al. (2014)), to further improve its convergence properties is a promising direction for future work. Currently, our method assumes that the tree structure is known, but in practice, the tree structure itself is not observed. The best tree structure may be obtained from the data by searching over all possible tree structures and choosing the one that results in the highest likelihood value. Such a search, though feasible in theory, is computationally challenging in practice because of the exponential number of possible trees. A potential research direction is to exploit linearization techniques and reduce the problem of joint estimation of tree structure and model parameters to a solving a sequence of mixed integer linear programs. Extending our method to allow for product features is also a natural next step.

References

Abdallah, Tarek, Gustavo Vulcano. 2021. Demand estimation under the multinomial logit model from sales transaction data. *Manufacturing & Service Operations Management* **23**(5) 1196–1216.

Akşin, Zeynep, Barış Ata, Seyed Morteza Emadi, Che-Lin Su. 2013. Structural estimation of callers' delay sensitivity in call centers. *Management Science* **59**(12) 2727–2746.

Allen-Zhu, Zeyuan, Lorenzo Orecchia. 2014. Linear coupling: An ultimate unification of gradient and mirror descent. *arXiv preprint arXiv:1407.1537* .

ALOGIT. 2018. The basic methods of ALOGIT. URL `www.alogit.com/Papers.htm`.

Anas, A. 1982. *Residential Location Markets and Urban Transportation: Economic Theory, Econometrics and Policy Analysis with Discrete Choice Models*. Academic Press, New York, NY.

Boyd, S., L. Vandenberghe. 2004. *Convex Optimization*. Cambridge University Press, Cambridge, UK.

Brownstone, D., K. A. Small. 1989. Efficient estimation of nested logit models. *Journal of Business and Economic Statistics* **7**(1) 67–74.

Bubeck, S. 2015. Convex optimization: Algorithms and complexity. *Foundations and Trends in Machine Learning* **8**(3-4) 231–357.

Cardell, N. S. 1997. Variance components structures for the extreme-value and logistic distributions with applications to models of heterogeneity. *Economic Theory* **13**(2) 185–213.

Daganzo, C. F., M. Kusnic. 1990. Large-scale nested logit models: Theory and experience. General Motors Economic and Marketing and Product Planning Staffs Report, Detroit, MI. (Abridged and updated version available as "Another look at the nested logit model," I.T.S. Research Report UCB-ITS-RR-92-2, UC Berkeley, 1992).

Daganzo, C. F., M. Kusnic. 1993. Two properties of the nested logit model. *Transportation Science* **27**(4) 395–400.

Daly, A. 1987. Estimating "tree" logit models. *Transportation Research Part B: Methodological* **21**(4) 251–267.

Davis, J. M., G. Gallego, H. Topaloglu. 2014. Assortment optimization under variants of the nested logit model. *Operations Research* **62**(2) 250–273.

Davis, J. M., H. Topaloglu, D. P. Williamson. 2016. Pricing problems under the nested logit model with a quality consistency constraint. *INFORMS Journal on Computing* **29**(1) 54–76.

Defazio, Aaron, Francis Bach, Simon Lacoste-Julien. 2014. Saga: A fast incremental gradient method with support for non-strongly convex composite objectives. *Advances in neural information processing systems* **27**.

Dempster, A. P., N. M. Laird, D. B. Rubin. 1977. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)* **39**(1) 1–38.

Dubé, Jean-Pierre, Jeremy T Fox, Che-Lin Su. 2012. Improving the numerical performance of static and dynamic aggregate discrete choice random coefficients demand estimation. *Econometrica* **80**(5) 2231–2267.

Feldman, J. B., H. Topaloglu. 2015. Capacity constraints across nests in assortment optimization under the nested logit model. *Operations Research* **63**(4) 812–822.

Forinash, C. V., F. S. Koppelman. 1993. Application and interpretation of nested logit models of intercity mode choice. *Transportation Research Record* (1413) 98–106.

Gallego, G., A. Li, V.-A. Truong, X. Wang. 2018. Online personalized resource allocation with customer choice. Working Paper, Columbia University.

Gallego, G., H. Topaloglu. 2014. Constrained assortment optimization for the nested logit model. *Management Science* **60**(10) 2583–2601.

Gallego, G., R. Wang. 2014. Multiproduct price optimization and competition under the nested logit model with product-differentiated price sensitivities. *Operations Research* **62**(2) 450–461.

Greene, W. H. 2018. NLOGIT Version 5 Reference Guide by Econometric Software. URL `www.scribd.com/document/355325579/NLOGIT-5-Reference-Guide`.

Gumbel, E. J. 2004. *Statistics of Extremes*. Dover Publications, Meneola, NY.

Hensher, D. A. 1986. Sequential and full information maximum likelihood estimation of a nested logit model. *Review of Economics and Statistics* **68**(4) 657–667.

Hensher, D. A., W. H. Greene. 2002. Specification and estimation of the nested logit model: Alternative normalisations. *Transportation Research Part B: Methodological* **36**(1) 1–17.

Hunt, Gary L. 2000. Alternative nested logit model structures and the special case of partial degeneracy. *Journal of Regional science* **40**(1) 89–113.

Hunter, D. R. 2004. MM algorithms for generalized Bradley-Terry models. *Annals of Statistics* **32**(1) 384–406.

Hunter, D. R., K. Lange. 2000. Rejoinder to discussion of "Optimization transfer using surrogate objective functions". *Journal of Computational and Graphical Statistics* **9**(1) 52–59.

Hunter, D. R., K. Lange. 2004. A tutorial on MM algorithms. *American Statistician* **58**(1) 30–37.

Johnson, Rie, Tong Zhang. 2013. Accelerating stochastic gradient descent using predictive variance reduction. *Advances in neural information processing systems* **26**.

Kamishima, Toshihiro. 2003. Nantonac collaborative filtering: recommendation based on order responses. *Proceedings of the ninth ACM SIGKDD international conference on Knowledge discovery and data mining*. 583–588.

Kiefer, J. 1953. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society* **4**(3) 502–506.

Kök, A. G., Y. Xu. 2011. Optimal and competitive assortments with endogenous pricing under hierarchical consumer choice models. *Management Science* **57**(9) 1546–1563.

Koppelman, F. S., C. Bhat. 2006. *A self instructing course in mode choice modeling: Multinomial and nested logit models*. Federal Transit Administration, 1200 New Jersey Avenue, SE, Washington, DC, United States 20590. Available at `www.ce.utexas.edu/prof/bhat/COURSES/LM_Draft_060131Final-060630.pdf`.

Koppelman, F. S., C.-H. Wen. 1998. Alternative nested logit models: Structure, properties and estimation. *Transportation Research Part B: Methodological* **32**(5) 289–298.

Lacoste-Julien, Simon. 2016. Convergence rate of frank-wolfe for non-convex objectives. *arXiv preprint arXiv:1607.00345* .

Lange, K., D. R. Hunter, I. Yang. 2000. Optimization transfer using surrogate objective functions. *Journal of Computational and Graphical Statistics* **9**(1) 1–20.

Lee, B. 1999. Calling patterns and usage of residential toll service under self selecting tariffs. *Journal of Regulatory Economics* **16**(1) 45–82.

Li, A. 2018. Product line design, pricing and framing under general choice models. Ph.D. thesis, Columbia University.

Li, G., P. Rusmevichientong, H. Topaloglu. 2015. The d-level nested logit model: Assortment and price optimization problems. *Operations Research* **63**(2) 325–342.

Li, H., W. T. Huh. 2011. Pricing multiple products with the multinomial logit and nested logit models: Concavity and implications. *Manufacturing & Service Operations Management* **13**(4) 549–563.

McFadden, D. 1978. Modeling the choice of residential location. *Transportation Research Record* (672) 72–77.

McFadden, D. 1981. Econometric models of probabilistic choice. *Structural Analysis of Discrete Data*. MIT Press, Cambridge, MA.

Mishra, V. K., K. Natarajan, D. Padmanabhan, C.-P. Teo, X. Li. 2014. On theoretical and empirical aspects of marginal distribution choice models. *Management Science* **60**(6) 1511–1531.

Nesterov, Y. 2013. *Introductory Lectures on Convex Optimization: A Basic Course*, vol. 87. Springer Science & Business Media.

Nocedal, J., S. J. Wright. 2006a. *Numerical Optimization*. 2nd ed. Springer series in Operations Research, Springer, New York, NY.

Nocedal, Jorge, Stephen Wright. 2006b. *Numerical optimization*. Springer Science & Business Media.

Rayfield, W. Z., P. Rusmevichientong, H. Topaloglu. 2015. Approximation methods for pricing problems under the nested logit model with price bounds. *INFORMS Journal on Computing* **27**(2) 335–357.

SAS. 2018. SAS/ETS 13.2 User's Guide: The MDC Procedure. URL `https://support.sas.com/documentation/onlinedoc/ets/132/mdc.pdf`.

Silberhorn, N., Y. Boztug, L. Hildebrandt. 2008. Estimation with the nested logit model: Specifications and software particularities. *OR Spectrum* **30**(4) 635.

Şimşek, A. S., H. Topaloglu. 2018. Technical note—an expectation-maximization algorithm to estimate the parameters of the markov chain choice model. *Operations Research* **66**(3) 748–760.

STATA. 2018a. Manual for MAXIMIZE subroutine. URL `www.stata.com/manuals/rmaximize.pdf`.

STATA. 2018b. Manual for NLOGIT subroutine. URL `www.stata.com/manuals/rnlogit.pdf`.

Tarjan, R. E. 1972. Depth-first search and linear graph algorithms. *SIAM Journal on Computing* **1**(2) 146–160.

Train, K. 1980. A structured logit model of auto ownership and mode choice. *Review of Economic Studies* **47**(2) 357–370.

Train, K. 1988. Qualitative choice analysis: Theory, econometrics, and an application to automobile demand. *Transportation Research* **22**(3) 233–235.

Train, K. 2009. *Discrete Choice Methods with Simulation*. 2nd ed. Cambridge University Press, Cambridge, UK.

Vojnovic, Milan, Se-Young Yun, Kaifang Zhou. 2020. Convergence rates of gradient descent and mm algorithms for bradley-terry models. *International Conference on Artificial Intelligence and Statistics*. PMLR, 1254–1264.

Vulcano, G., G. J. van Ryzin, R. Ratliff. 2012. Estimating primary demand for substitutable products from sales transaction data. *Operations Research* **60**(2) 313–334.

Zachary, Stan. 1978. Improved multiple choice models. *Hensher DA* 335–357.

Zhang, Xiao-Dong. 2011. The laplacian eigenvalues of graphs: a survey. *arXiv preprint arXiv:1111.2897* .

This page is intentionally blank. Proper e-companion title page, with INFORMS branding and exact metadata of the main paper, will be produced by the INFORMS office when the issue is being assembled.

Online Appendix

Estimating Large-Scale Tree Logit Models

Appendix A: Proof of Theorem 2.1 (Random Utility Representation)

Recall that a random variable x follows a Gumbel distribution with a location parameter μ and a scaling parameter β if for all $x \in \mathbb{R}$, $\mathbb{P}\{x \leq x\} = e^{-e^{-(x-\mu)/\beta}}$, and we denote this by $x \sim \mathsf{Gumbel}(\mu, \beta)$. We say that x follows a *standard* Gumbel distribution if $x \sim \mathsf{Gumbel}(0, 1)$. The proof of Theorem 2.1 makes use of the following standard results about the Gumbel distribution (Gumbel 2004).

Lemma A.1 (Gumbel Properties) *Suppose $x \sim \mathsf{Gumbel}(\mu, \beta)$, and let $x_1, \ldots, x_n$ be independent Gumbel random variables with $x_j \sim \mathsf{Gumbel}(\mu_j, \beta)$ for all j. Then,*

1. $\mathbb{E}[x] = \mu + \beta\gamma$ *where* $\gamma = 0.57721...$ *is the Euler-Mascheroni constant and* $\mathsf{Var}[x] = \pi^2\beta^2/6$.
2. *For any* $b > 0$ *and* $a \in \mathbb{R}$, $bx + a \sim \mathsf{Gumbel}(b\mu + a, b\beta)$.
3. *The random variable $x_1 - x_2$ follows a Logistic distribution with a location parameter $\mu_1 - \mu_2$ and scale parameter β; that is, for all $x \in \mathbb{R}$, $\mathbb{P}\{x_1 - x_2 \leq x\} = \frac{1}{1+e^{-[x-(\mu_1-\mu_2)]/\beta}}$.*
4. *The random variable $\max_{i=1,\ldots,n} x_j$ follows Gumbel distribution with location parameter $\beta \ln\left(\sum_{j=1}^n e^{\mu_j/\beta}\right)$ and scale parameter β, and $\mathbb{P}\{x_j > \max_{\ell \neq j} x_\ell\} = \frac{e^{\mu_j/\beta}}{\sum_{\ell=1}^n e^{\mu_\ell/\beta}}$.*
5. *The random variable $\max\{x_1, x_2\}$ is independent of $\mathbb{1}\{x_1 > x_2\}$.*

The next lemma shows that there exists a unique distribution such that when a random variable following this distribution is added to an independent $\mathsf{Gumbel}(0, \beta)$ random variable, with $\beta < 1$, the resulting sum is a standard Gumbel random variable.

Lemma A.2 (Theorem 2.1 in Cardell 1997) *Suppose $x \sim \mathsf{Gumbel}(0, \beta)$ with $0 < \beta < 1$ and there is another random variable y independent of x. Then, $x + y$ has a standard Gumbel distribution if and only if y has a density function $f_\beta(y) = \frac{1}{\beta}\sum_{k=0}^{\infty} \frac{(-1)^k e^{-ky}}{k!\Gamma(-\beta k)}$ for all $y \in \mathbb{R}$.*

As an immediate corollary, there exists an independent random variable that can be added to another Gumbel random variable to obtain a Gumbel distribution with a higher scaling parameter.

Corollary A.3 (Changing scale through addition) *Suppose $x \sim \mathsf{Gumbel}(\mu, \nu)$, with $\mu \in \mathbb{R}$ and $\nu > 0$, and we are given $\lambda > 0$ such that $\lambda \geq \nu$. Then, there exists a random variable y such that y is independent of x and $x + y \sim \mathsf{Gumbel}(\mu, \lambda)$.*

The requirement that $\nu \leq \lambda$ in the above corollary is necessary. Since x and y are independent, we have that $\mathsf{Var}(x+y) = \mathsf{Var}(x) + \mathsf{Var}(y)$, which implies that $\pi^2\lambda^2/6 = \pi^2\nu^2/6 + \mathsf{Var}(y)$. Because the variance of a random variable is always non-negative, it must always be true that $\lambda \geq \nu$.

For each leaf node $\ell \in \mathcal{N}$ and each node j that is an ancestor of ℓ, recall that $\mathsf{path}(j, \ell]$ denotes the nodes on the unique path from j to ℓ, excluding j. Define

$$z_{j,\ell} = v_j + \sum_{k \in \mathsf{path}(j,\ell]} v_k \; + \; \mu_\ell.$$

Since $v_{\mathsf{root}} = 0$, we have $\mathit{utility}_\ell = z_{\mathsf{root},\ell}$. The following lemma describes the distribution of $z_{j,\ell}$.

Lemma A.4 *For each leaf node ℓ and its ancestor j such that $j \neq \mathsf{root}$, $z_{j,\ell} \sim \mathsf{Gumbel}\left(\mu_\ell, \lambda_{\mathsf{pa}(j)}\right)$.*

Proof: Fix an arbitrary leaf node ℓ. We prove this result by induction on the height[11] of the ancestor j. The base case is when the height of j is zero, which means that $j = \ell$. By definition, $z_{j,\ell} = v_\ell + \mu_\ell \; \sim \; \mathsf{Gumbel}\left(\mu_\ell, \lambda_{\mathsf{pa}(j)}\right)$, which is the desired result.

To establish the induction step, we assume that $z_{j,\ell} \sim \mathsf{Gumbel}(\mu_\ell, \lambda_{\mathsf{pa}(j)})$ for all ancestors j of ℓ whose heights are at most H. Consider an ancestor j with height $H+1$. Let k denote the child of j that is also an ancestor of ℓ; note that the height of k is at most H. By definition, $z_{j,\ell} = v_j + z_{k,\ell}$. Since k has a height of H, it follows from the induction hypothesis that $z_{k,\ell} \sim \mathsf{Gumbel}(\mu_\ell, \lambda_{\mathsf{pa}(k)})$. As j is the parent of k, we have that $\lambda_{\mathsf{pa}(k)} = \lambda_j$ and $z_{k,\ell} \sim \mathsf{Gumbel}(\mu_\ell, \lambda_j)$. The definition of v_j then implies that $z_{j,\ell} = v_j + z_{k,\ell} \sim \mathsf{Gumbel}(\mu_\ell, \lambda_{\mathsf{pa}(j)})$, completing the induction step. This completes the proof. ■

For each $\mathcal{S} \subseteq \mathcal{N}$ and each node $j \in \mathsf{T}[\mathcal{S}]$, let the $Z_j(\mathcal{S})$ denote the maximum of random variables $z_{j,\ell}$ over all the leaf nodes in $\mathsf{T}[\mathcal{S}]$ that are descendants of j; that is,

$$Z_j(\mathcal{S}) \; \stackrel{\text{def}}{=} \; \max\{z_{j,\ell} \colon \ell \text{ is a leaf node in } \mathsf{T}_j[\mathcal{S}]\} \;\; ,$$

where we define the maximum over an empty set to be minus infinity. The next lemma characterizes the distribution of $Z_j(\mathcal{S})$.

Lemma A.5 *For each subset $\mathcal{S} \subseteq \mathcal{N}$ and each node $j \in \mathsf{T}[\mathcal{S}]$ such that $j \neq \mathsf{root}$, $Z_j(\mathcal{S}) \; \sim \; \mathsf{Gumbel}\left(W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \; , \; \lambda_{\mathsf{pa}(j)}\right)$.*

Proof: Without loss of generality, we prove the result for the case $\mathcal{S} = \mathcal{N}$, so $\mathsf{T}[\mathcal{S}] = \mathsf{T}$. The proof for a general subset $\mathcal{S}$ follows from an identical argument applied on the sub-tree $\mathsf{T}[\mathcal{S}]$. Since we consider the full assortment, we will drop references to $\mathcal{S}$, and simply write Z_j and T_j. We will

[11] Recall that the height of a node is the number of edges on the longest path between that node and a leaf.

prove the result by induction on the height of node j. For the base case, suppose that the height of j is zero. So, j is a leaf node. Then,

$$Z_j = \max\{z_{j,\ell} \colon \ell \text{ is a leaf node in } \mathsf{T}_j\} = z_{j,j} = \mu_j + v_j \;\sim\; \mathsf{Gumbel}\left(W_j(\boldsymbol{\mu},\boldsymbol{\lambda}) \;,\; \lambda_{\mathsf{pa}(j)}\right) \;,$$

where we use the fact that if j is a leaf node, then $v_j \sim \mathsf{Gumbel}\left(0, \lambda_{\mathsf{pa}(j)}\right)$ and $W_j(\boldsymbol{\mu},\boldsymbol{\lambda}) = \mu_j$. This completes the base case.

To establish the induction step, we assume that the result holds for all nodes with height of at most H. We now consider node j with height of $H+1$. Since $Z_j = \max\{z_{j,\ell} \colon \ell \text{ is a leaf node in } \mathsf{T}_j\}$, it follows that that

$$Z_j \;=\; v_j + \max_{k \in \mathsf{Children}(j)} \left\{ \max\{z_{k,\ell} \colon \ell \text{ is a leaf node in } \mathsf{T}_k\} \right\} \;=\; v_j + \max_{k \in \mathsf{Children}(k)} Z_k.$$

For each $k \in \mathsf{Children}(j)$, the height of k is at most H. So, invoking the induction hypothesis, we obtain that for all $k \in \mathsf{Children}(j)$,

$$Z_k \;=\; \max\{z_{k,\ell} \colon \ell \text{ is a leaf node in } \mathsf{T}_k\} \;\sim\; \mathsf{Gumbel}\left(W_k(\boldsymbol{\mu},\boldsymbol{\lambda}), \lambda_j\right)$$

because $\lambda_{\mathsf{pa}(k)} = \lambda_j$. Further, because the vertex sets of T_k and $\mathsf{T}_{k'}$ are disjoint for any $k \neq k'$ such that $\{k, k'\} \subseteq \mathsf{Children}(j)$, we have that Z_k is independent of $Z_{k'}$. It then follows from the properties of the Gumbel distribution that

$$\max_{k \in \mathsf{Children}(k)} Z_k \sim \mathsf{Gumbel}\left(\lambda_j \log\left(\sum_{k \in \mathsf{Children}(j)} e^{W_k(\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}\right) \;,\; \lambda_j\right) = \mathsf{Gumbel}\left(W_j(\boldsymbol{\mu},\boldsymbol{\lambda}) \;,\; \lambda_j\right),$$

where the equality follows from the definition of $W_j(\boldsymbol{\mu},\boldsymbol{\lambda})$. It thus follows from the defition of v_j that $Z_j \sim \mathsf{Gumbel}(W_j(\boldsymbol{\mu},\boldsymbol{\lambda}), \lambda_{\mathsf{pa}(j)})$. This establishes the induction step, completing the proof. ■

The next lemma allows us to simplify the conditional probability involving $Z_j(\mathcal{S})$.

Lemma A.6 *For each subset $\mathcal{S} \subseteq \mathcal{N}$ and each node $j \in \mathsf{T}[\mathcal{S}]$ such that $j \neq \mathsf{root}$,*

$$\begin{aligned}\mathbb{P}\left\{ Z_j(\mathcal{S}) \;>\; \max_{k \in \mathsf{Sibling}(j)\setminus\{j\}} Z_k(\mathcal{S}) \;\middle|\; Z_i(\mathcal{S}) \;>\; \max_{v \in \mathsf{Sibling}(i)\setminus\{i\}} Z_v(\mathcal{S}) \;\;\forall\; i \in \mathsf{path}\left(\mathsf{root}, \mathsf{pa}(j)\right)\right\} \\ =\; \mathbb{P}\left\{ Z_j(\mathcal{S}) \;>\; \max_{k \in \mathsf{Sibling}(j)\setminus\{j\}} Z_k(\mathcal{S}) \right\} \;,\end{aligned}$$

where $\mathsf{Sibling}(j) = \{k \in \mathsf{T}[\mathcal{S}] \;:\; \mathsf{pa}(k) = \mathsf{pa}(j)\}$ denote the siblings of j in the tree $\mathsf{T}[\mathcal{S}]$.

Proof: Without loss of generality, assume that $\mathcal{S} = \mathcal{N}$. The proof for the general $\mathcal{S}$ is essentially the same. Since we consider the full assortment, we will drop references to $\mathcal{S}$. Fix an arbitrary node $j \in \mathsf{T}$ such that $j \neq \mathsf{root}$. Let $x_1 = Z_j$ and $x_2 = \max_{k \in \mathsf{Sibling}(j)\setminus\{j\}} Z_k$. Note that x_1 and x_2 are independent of each other. We will first establish the following claim by induction.

Claim: For every node $i \in \mathsf{path}(\mathsf{root}, \mathsf{pa}(j)]$, there is a deterministic function f_i such that $Z_i = f_i\left(\max\{x_1, x_2\}, \boldsymbol{U}_i\right)$, where the random vector $\boldsymbol{U}_i$ is independent of x_1 and x_2.

We will prove the claim by induction on node i. For the base case, consider $i = \mathsf{pa}(j)$. Then, by definition,

$$Z_{\mathsf{pa}(j)} \;=\; v_{\mathsf{pa}(j)} + \max_{k \in \mathsf{Children}(\mathsf{pa}(j))} Z_k \;=\; v_{\mathsf{pa}(j)} + \max\left\{ Z_j \,,\, \max_{k \in \mathsf{Sibling}(j) \setminus \{j\}} Z_k \right\} \;=\; v_{\mathsf{pa}(j)} + \max\{x_1, x_2\} \;,$$

and the result follows because $v_{\mathsf{pa}(j)}$ is independent of x_1 and x_2. This proves the base case.

For the induction step, assume the result holds for some node $i \in \mathsf{path}(\mathsf{root}, \mathsf{pa}(j)]$. We will now prove that it also holds for node $\mathsf{pa}(i)$. By definition,

$$\begin{aligned} Z_{\mathsf{pa}(i)} \;&=\; v_{\mathsf{pa}(i)} + \max_{v \in \mathsf{Children}(\mathsf{pa}(i))} Z_v \;=\; v_{\mathsf{pa}(i)} + \max\left\{ Z_i \,,\, \max_{v \in \mathsf{Sibling}(i) \setminus \{i\}} Z_v \right\} \\ &=\; v_{\mathsf{pa}(i)} + \max\left\{ f_i\left(\max\{x_1, x_2\}, \boldsymbol{U}_i\right) \,,\, \max_{v \in \mathsf{Sibling}(i) \setminus \{i\}} Z_v \right\} \end{aligned}$$

and the desired result follows because $v_{\mathsf{pa}(i)}$ is independent of x_1, x_2, and $\boldsymbol{U}_i$. Moreover, $\max_{v \in \mathsf{Sibling}(i) \setminus \{i\}} Z_v$ is also independent of x_1 and x_2 because for each $v \in \mathsf{Sibling}(i) \setminus \{i\}$, the sub-tree T_v rooted at v is completely separate from the sub-tree T_k for all $k \in \mathsf{Sibling}(j)$. This completes the induction, establishing the claim.

It follows from the above claim that

$$\begin{aligned} &\mathbb{P}\left\{ Z_j \;>\; \max_{k \in \mathsf{Sibling}(j) \setminus \{j\}} Z_k \;\middle|\; Z_i \;>\; \max_{v \in \mathsf{Sibling}(i) \setminus \{i\}} Z_v \;\; \forall\, i \in \mathsf{path}\left(\mathsf{root}, \mathsf{pa}(j)\right] \right\} \\ &\quad= \mathbb{P}\left\{ x_1 \;>\; x_2 \;\middle|\; f_i\left(\max\{x_1, x_2\}, \boldsymbol{U}_i\right) \;>\; \max_{v \in \mathsf{Sibling}(i) \setminus \{i\}} Z_v \;\; \forall\, i \in \mathsf{path}\left(\mathsf{root}, \mathsf{pa}(j)\right] \right\} \\ &\quad= \mathbb{P}\{x_1 \;>\; x_2\} \;, \end{aligned}$$

where the last equality follows from Lemma A.1, which shows that $\mathbb{1}\{x_1 > x_2\}$ is independent of $\max\{x_1, x_2\}$. Also, note that for all $i \in \mathsf{path}\left(\mathsf{root}, \mathsf{pa}(j)\right]$, $\max_{v \in \mathsf{Sibling}(i) \setminus \{i\}} Z_v$ is independent of x_1 and x_2. This completes the induction, proving the desired result.

Finally, here is the proof of Theorem 2.1.

Proof of Theorem 2.1: Fix an arbitrary subset $\mathcal{S}$ and $\ell \in \mathcal{S}$. Recall that $\mathsf{Sibling}(j) = \{k \in \mathsf{T}[\mathcal{S}] \;:\; \mathsf{pa}(k) = \mathsf{pa}(j)\}$ denote the siblings of j in the tree $\mathsf{T}[\mathcal{S}]$. Let $\mathsf{root} \to j_1 \to j_2 \to \cdots \to j_m \to \ell$ denote the unique path from root to ℓ in $\mathsf{T}[\mathcal{S}]$. Note that

$$\{\ell\} \;=\; \mathsf{T}_\ell[\mathcal{S}] \cap \mathcal{S} \;\subseteq\; \mathsf{T}_{j_m}[\mathcal{S}] \cap \mathcal{S} \;\subseteq\; \mathsf{T}_{j_{m-1}}[\mathcal{S}] \cap \mathcal{S} \;\cdots\; \subseteq\; \mathsf{T}_{j_1}[\mathcal{S}] \cap \mathcal{S} \;\subseteq\; \mathsf{T}_{\mathsf{root}}[\mathcal{S}] \cap \mathcal{S} = \mathcal{S}$$

Note that the event $\mathit{utility}_\ell > \max_{k \in \mathcal{S} \setminus \{\ell\}} \mathit{utility}_k$ happens if and only if for every node $j \in \mathsf{path}(\mathsf{root}, \ell]$, we have $\max_{k \in \mathsf{T}_j[\mathcal{S}] \cap \mathcal{S}} \mathit{utility}_k \;>\; \max_{i \in \mathsf{Sibling}(j) \setminus \{j\}} \max_{k \in \mathsf{T}_i[\mathcal{S}] \cap \mathcal{S}} \mathit{utility}_k$. Therefore,

$$\mathbb{P}\left\{ \mathit{utility}_\ell > \max_{k \in \mathcal{S} \setminus \{\ell\}} \mathit{utility}_k \right\}$$

$$= \mathbb{P}\left\{ \max_{k \in \mathsf{T}_j[\mathcal{S}] \cap \mathcal{S}} \mathit{utility}_k \; > \max_{i \in \mathsf{Sibling}(j) \setminus \{j\}} \max_{k \in \mathsf{T}_i[\mathcal{S}] \cap \mathcal{S}} \mathit{utility}_k \quad \forall\, j \in \mathsf{path}(\mathsf{root}, \ell] \right\}$$
$$= \mathbb{P}\left\{ Z_j(\mathcal{S}) \; > \max_{i \in \mathsf{Sibling}(j) \setminus \{j\}} Z_i(\mathcal{S}) \quad \forall\, j \in \mathsf{path}(\mathsf{root}, \ell] \right\} ,$$

where the last equality follows because for each $i \in \mathsf{Sibling}(j)$

$$\max_{k \in \mathsf{T}_i[\mathcal{S}] \cap \mathcal{S}} \mathit{utility}_k \; = \sum_{v \in \mathsf{path}(\mathsf{root}, \mathsf{pa}(j)]} v_v \; + \max_{k \in \mathsf{T}_i[\mathcal{S}] \cap \mathcal{S}} z_{i,k} \; = \sum_{v \in \mathsf{path}(\mathsf{root}, \mathsf{pa}(j)]} v_v \; + \; Z_i(\mathcal{S}) \; ,$$

and the term $\sum_{v \in \mathsf{path}(\mathsf{root}, \mathsf{pa}(j)]} v_v$ is common for all nodes $i \in \mathsf{Sibling}(j)$.

For any collection of random variables $x_1, \ldots, x_n$, $\mathbb{P}\{x_1, \ldots, x_n\} = \prod_{j=1}^{n} \mathbb{P}\left\{ x_j \;\middle|\; x_{j-1}, \ldots, x_1 \right\}$. Thus,

$$\mathbb{P}\left\{ Z_j(\mathcal{S}) \; > \max_{i \in \mathsf{Sibling}(j) \setminus \{j\}} Z_i(\mathcal{S}) \quad \forall\, j \in \mathsf{path}(\mathsf{root}, \ell] \right\}$$
$$= \prod_{j \in \mathsf{path}(\mathsf{root}, \ell]} \mathbb{P}\left\{ Z_j(\mathcal{S}) \; > \max_{i \in \mathsf{Sibling}(j) \setminus \{j\}} Z_i(\mathcal{S}) \;\middle|\; Z_k > \max_{v \in \mathsf{Sibling}(k) \setminus \{k\}} Z_k \quad \forall\, k \in \mathsf{path}(\mathsf{root}, \mathsf{pa}(j)] \right\}$$
$$= \prod_{j \in \mathsf{path}(\mathsf{root}, \ell]} \mathbb{P}\left\{ Z_j(\mathcal{S}) \; > \max_{i \in \mathsf{Sibling}(j) \setminus \{j\}} Z_i(\mathcal{S}) \right\}$$

where last equality follows from Lemma A.6.

By Lemma A.5, $Z_j(\mathcal{S}) \sim \mathsf{Gumbel}\left(W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \; , \; \lambda_{\mathsf{pa}(j)} \right)$, and for each $i \in \mathsf{Sibling}(j) \setminus \{j\}$, $Z_i(\mathcal{S}) \sim \mathsf{Gumbel}\left(W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \; , \; \lambda_{\mathsf{pa}(i)} \right)$. Since $\lambda_{\mathsf{pa}(i)} = \lambda_{\mathsf{pa}(j)}$ for all $i \in \mathsf{Sibling}(j)$,

$$\mathbb{P}\left\{ Z_j(\mathcal{S}) \; > \max_{i \in \mathsf{Sibling}(j) \setminus \{j\}} Z_i(\mathcal{S}) \right\} = \frac{e^{W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) / \lambda_{\mathsf{pa}(j)}}}{\sum_{i \in \mathsf{Sibling}(j)} e^{W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) / \lambda_{\mathsf{pa}(j)}}} \; ,$$

which implies that

$$\mathbb{P}\left\{ \mathit{utility}_\ell > \max_{k \in \mathcal{S} \setminus \{\ell\}} \mathit{utility}_k \right\} \; = \prod_{j \in \mathsf{path}(\mathsf{root}, \ell]} \frac{e^{W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) / \lambda_{\mathsf{pa}(j)}}}{\sum_{i \in \mathsf{Sibling}(j)} e^{W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) / \lambda_{\mathsf{pa}(j)}}} \; = \; \psi_{\mathsf{root} \to \ell}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) = \mathbb{P}_\ell\left(\mathcal{S} \,;\, \boldsymbol{\mu}, \boldsymbol{\lambda} \right) \; ,$$

and this completes the proof. ■

Appendix B: Properties of the Weight Function

The following lemmas establish important properties of the weight function $W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$, which we will use repeatedly to establish properties of the negative log-likelihood function. We first state all the lemmas and then provide their proofs. The first lemma establishes positive homogeneity and additivity.

Lemma B.1 (Positive Homogeneity and Additivity) *For each subset $\mathcal{S} \subseteq \mathcal{N}$, node $j \in \mathsf{T}[\mathcal{S}]$, $\alpha > 0$, and $\xi \in \mathbb{R}$, $W_j(\mathcal{S}; \alpha\boldsymbol{\mu}, \alpha\boldsymbol{\lambda}) = \alpha W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ and $W_j(\mathcal{S}; \boldsymbol{\mu} + \xi \boldsymbol{e}, \boldsymbol{\lambda}) = \xi + W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$, where $\boldsymbol{e}$ is the vector of all ones.*

The next lemma establishes convexity, and under additional assumptions, strict convexity. Recall that for each node $j \in \mathsf{T}[\mathcal{S}]$, $\mathsf{T}_j[\mathcal{S}]$ denotes the sub-tree of $\mathsf{T}[\mathcal{S}]$ rooted at j; that is, $\mathsf{T}_j[\mathcal{S}]$ consists of the node j and all of its descendant in $\mathsf{T}[\mathcal{S}]$.

Lemma B.2 (Convexity and Strict Convexity) *For each subset $\mathcal{S} \subseteq \mathcal{N}$ and $j \in \mathsf{T}[\mathcal{S}]$, the function $(\boldsymbol{\mu}, \boldsymbol{\lambda}) \mapsto W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ satisfies the following properties:*

(a) *It is convex in $(\boldsymbol{\mu}, \boldsymbol{\lambda})$.*

(b) *For each $\boldsymbol{\lambda}$ and $0 < \theta < 1$,*

$$W_j(\mathcal{S}; \theta\boldsymbol{\mu} + (1-\theta)\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) = \theta W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) + (1-\theta) W_j(\mathcal{S}; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})$$

if and only if there exists $\xi \in \mathbb{R}$ such that $\bar{\mu}_\ell = \mu_\ell + \xi$ for all leaf nodes $\ell \in \mathsf{T}_j[\mathcal{S}] \cap \mathcal{S}$.

(c) *For each $a \in \mathbb{R}$ and $\boldsymbol{\lambda}$, the function $\boldsymbol{\mu} \mapsto W_{\mathsf{root}}(\mathcal{N}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ is strictly convex on the set $\{\boldsymbol{\mu} : \mu_1 = a\}$.*

The next lemma establishes the monotonicity of the weight function.

Lemma B.3 (Monotonicity and Strict Monotonicity) *For each $\mathcal{S} \subseteq \mathcal{N}$ and $j \in \mathsf{T}[\mathcal{S}]$, the function $(\boldsymbol{\mu}, \boldsymbol{\lambda}) \mapsto W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ satisfies the following properties.*

(a) *It is increasing in $(\boldsymbol{\mu}, \boldsymbol{\lambda})$; that is, if $(\boldsymbol{\mu}, \boldsymbol{\lambda}) \leq (\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}})$ where the inequality holds componentwise, then $W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \leq W_j(\mathcal{S}; \bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}})$.*

(b) *It is strictly increasing in μ_ℓ for each leaf node $\ell \in \mathsf{T}_j[\mathcal{S}] \cap \mathcal{S}$.*

(c) *It is strictly increasing in λ_k for each $k \in \mathsf{T}_j[\mathcal{S}]$ such that k has at least two children in $\mathsf{T}_j[\mathcal{S}]$.*

Finally, the following lemma provides an expression for the derivative of the weight function with respect to the model parameters.

Lemma B.4 (Derivatives of the Weight Functions) *For each $\mathcal{S} \subseteq \mathcal{N}$ and $j \in \mathsf{T}[\mathcal{S}]$,*

$$\begin{aligned}
\frac{\partial W_j}{\partial \mu_\ell}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) &= \psi_{j \to \ell}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) && \forall\ \ell \in \mathcal{N} \\
\frac{\partial W_j}{\partial \lambda_k}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) &= \psi_{j \to k}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})\, \Delta_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) && \forall\ k \in \mathsf{T} \setminus \mathcal{N},\ k \neq \mathsf{root}\ .
\end{aligned}$$

where for each non-leaf node k such that $k \neq \mathsf{root}$,

$$\begin{aligned}
\Delta_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) &= \log\left(\sum_{i \in \mathsf{Children}(k) \cap \mathsf{T}[\mathcal{S}]} e^{W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_k}\right) - \frac{\sum_{i \in \mathsf{Children}(k) \cap \mathsf{T}[\mathcal{S}]} e^{W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_k} \times W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_k \sum_{i \in \mathsf{Children}(k) \cap \mathsf{T}[\mathcal{S}]} e^{W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_k}} \\
&= \frac{W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) - \sum_{i \in \mathsf{Children}(k) \cap \mathsf{T}[\mathcal{S}]} \psi_{k \to i}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \times W_i(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_k}\ .
\end{aligned}$$

We now present the proofs of these four lemmas.

B.1 Proof of Lemma B.1

Proof: Consider an arbitrary subset $\mathcal{S}$. We prove both results by induction on the height of node j. For the base case, suppose the height of j is zero. This means that j is a leaf node of $\mathsf{T}[S]$, so $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \mu_j$ and the result is trivially true. Suppose that the results holds for all nodes at height at most H. Now consider a vertex $j \in \mathsf{T}[\mathcal{S}]$ of height $H+1$. By the induction hypothesis, the results are true for all children k of node j because the height of k is at most H. Therefore, by definition of W_j and the inductive hypothesis,

$$W_j(\mathcal{S};\alpha\boldsymbol{\mu},\alpha\boldsymbol{\lambda}) \;=\; \alpha\lambda_j \log\left(\sum_{k\in\,\mathsf{Children}(j)\,\cap\,\mathsf{T}[\mathcal{S}]} e^{\alpha W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/(\alpha\lambda_j)}\right) = \alpha W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \text{ and}$$

$$W_j(\mathcal{S};\boldsymbol{\mu}+\xi\boldsymbol{e},\boldsymbol{\lambda}) \;=\; \lambda_j \log\left(\sum_{k\in\,\mathsf{Children}(j)\,\cap\,\mathsf{T}[\mathcal{S}]} e^{[\xi+W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})]/\lambda_j}\right) = \xi + W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}).$$

We have thus established the result for nodes at height $H+1$, completing the induction step. The result of the lemma now follows. ■

B.2 Proof of Lemma B.2

Proof: Fix an arbitrary subset $\mathcal{S}$. We prove each of the three parts separately. Throughout this proof, we make use of the following observation: for each $j \in \mathsf{T}[\mathcal{S}]$, the function $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ only depends the parameters associated with nodes in the sub-tree $\mathsf{T}_j[\mathcal{S}]$ rooted at j; that is, $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \;=\; W_j\left(\mathcal{S};\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]},\boldsymbol{\lambda}_{\mathsf{T}_j[\mathcal{S}]}\right)$, where $\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]} = (\mu_\ell : \ell \in \mathsf{T}_j[\mathcal{S}])$ and $\boldsymbol{\lambda}_{\mathsf{T}_j[\mathcal{S}]} = (\lambda_k : k \in \mathsf{T}_j[\mathcal{S}])$.

Proof of part (a): We use induction on the height of node j. For the base case, suppose that the height of j is zero, so j is a leaf node. Then, the result is trivially true by definition. Suppose that $W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is convex in $(\boldsymbol{\mu},\boldsymbol{\lambda})$ for all nodes k with height at most H. Now, consider an arbitrary non-leaf node j at height $H+1$. By the induction hypothesis, $W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is convex in $(\boldsymbol{\mu},\boldsymbol{\lambda})$ for all children k of node j. Then, the function

$$(\boldsymbol{\mu},\boldsymbol{\lambda}) \mapsto \log\left(\sum_{k\in\,\mathsf{Children}(j)\,\cap\,\mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}\right)$$

must also be convex in $(\boldsymbol{\mu},\boldsymbol{\lambda})$ because it is a composition of the log-sum-exp function, which is increasing and convex, with a collection of convex functions $(\boldsymbol{\mu},\boldsymbol{\lambda}) \mapsto W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$, for $k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]$. Note that for each $k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]$, the function $W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is *independent* of λ_j because $j \notin \mathsf{T}_k[\mathcal{S}]$, so

$$\log\left(\sum_{k\in\,\mathsf{Children}(j)\,\cap\,\mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}\right) = \log\left(\sum_{k\in\,\mathsf{Children}(j)\,\cap\,\mathsf{T}[\mathcal{S}]} e^{W_k\left(\mathcal{S};\boldsymbol{\mu}_{\mathsf{T}_k[\mathcal{S}]},\boldsymbol{\lambda}_{\mathsf{T}_k[\mathcal{S}]}\right)}\right) .$$

Therefore, the perspective of the above function is given by

$$(\boldsymbol{\mu},\boldsymbol{\lambda}) \mapsto \lambda_j \log\left(\sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}[\mathcal{S}]} e^{W_k\left(\mathcal{S};\frac{\boldsymbol{\mu}_{\mathsf{T}_k[\mathcal{S}]}}{\lambda_j},\frac{\boldsymbol{\lambda}_{\mathsf{T}_k[\mathcal{S}]}}{\lambda_j}\right)}\right) = \lambda_j \log\left(\sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}[\mathcal{S}]} e^{W_k\left(\mathcal{S};\boldsymbol{\mu}_{\mathsf{T}_k[\mathcal{S}]},\boldsymbol{\lambda}_{\mathsf{T}_k[\mathcal{S}]}\right)/\lambda_j}\right)$$
$$= W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \ ,$$

where the first equality follows from the positive homogeneity property in Lemma B.1. Because the perspective of a convex function is also convex (Boyd and Vandenberghe 2004), it follows that $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is convex in $(\boldsymbol{\mu},\boldsymbol{\lambda})$, completing the induction step. This proves part (a).

Proof of part (b): We prove part (b) by induction on the height of node j. Consider the base case where j has a height of zero, so $j=\ell$ for some leaf node ℓ in $\mathsf{T}[\mathcal{S}]$. In this case, $\mathsf{T}_\ell[\mathcal{S}]=\{\ell\}$ and $W_\ell(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})=\mu_\ell$. Because $\mathsf{T}_\ell[\mathcal{S}]\cap\mathcal{S}$ contains only the node ℓ in it, the condition $\bar{\mu}_\ell=\mu_\ell+\xi$ can be trivially satisfied by choosing $\xi=\bar{\mu}_\ell-\mu_\ell$. Further, the relationship

$$W_\ell(\mathcal{S};\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}) = \theta W_\ell(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) + (1-\theta) W_\ell(\mathcal{S};\bar{\boldsymbol{\mu}},\boldsymbol{\lambda})$$

is also trivially satisfied by defnition for all $\theta\in(0,1)$. Therefore, the base case is true.

Suppose that the result holds for all nodes with height at most H. Consider an arbitrary non-leaf node j at height $H+1$. By the induction hypothesis, we have that for each $k\in\mathsf{Children}(j)$,

$$W_k(\mathcal{S};\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}) = \theta W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) + (1-\theta) W_k(\mathcal{S};\bar{\boldsymbol{\mu}},\boldsymbol{\lambda})$$

if and only if there exists $\xi_k\in\mathbb{R}$ such that $\mu_\ell=\bar{\mu}_\ell+\xi_k$ for all $\ell\in\mathsf{T}_k[\mathcal{S}]\cap\mathcal{S}$. We will now prove the result at node j.

We will first prove the sufficiency. Suppose that there exists $\xi\in\mathbb{R}$ such that $\bar{\mu}_\ell=\bar{\mu}_\ell+\xi$ for all leaf nodes $\ell\in\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}$, or equivalently $\bar{\boldsymbol{\mu}}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}}=\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}}+\xi\boldsymbol{e}$, where is $\boldsymbol{e}$ is a vector of ones of an appropriate dimension. Then,

$$\begin{aligned}
W_j(\mathcal{S};\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}) &= W_j(\mathcal{S};\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]}+\xi(1-\theta)\boldsymbol{e},\boldsymbol{\lambda}) && [W_j \text{ only depends on } \boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]}]\\
&= \xi(1-\theta)+W_j(\mathcal{S};\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]},\boldsymbol{\lambda}) && [\text{by Lemma } B.1]\\
&= \theta W_j(\mathcal{S};\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]},\boldsymbol{\lambda}) + (1-\theta)\left(\xi+W_j(\mathcal{S};\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]},\boldsymbol{\lambda})\right)\\
&= \theta W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) + (1-\theta) W_j(\mathcal{S};\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}),
\end{aligned}$$

where the last equality follows because $\bar{\boldsymbol{\mu}}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}}=\boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}}+\xi\boldsymbol{e}$. This gives the desired result.

We will now prove the necessity. Suppose that $\boldsymbol{\mu}$, $\bar{\boldsymbol{\mu}}$, and $\theta\in(0,1)$ are such that

$$W_j(\mathcal{S};\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}) = \theta W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) + (1-\theta) W_j(\mathcal{S};\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}). \tag{7}$$

Our goal is to exhibit a $\xi \in \mathbb{R}$ such that $\bar{\boldsymbol{\mu}}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}} + \xi = \boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}}$. If we have that $\bar{\boldsymbol{\mu}}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}} = \boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}}$, then the result is trivially true. Therefore, we assume that $\bar{\boldsymbol{\mu}}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}} \neq \boldsymbol{\mu}_{\mathsf{T}_j[\mathcal{S}]\cap\mathcal{S}}$. To simplify notation, let $a = 1/\lambda_j$, $m = |\,\mathsf{Children}(j)|$. Further, let $\mathsf{LSE}\colon \mathbb{R}^m \to \mathbb{R}_{++}$ be defined as $\mathsf{LSE}(\boldsymbol{x}) = \log\left(\sum_{i=1}^m e^{x_i}\right)$, for each $\boldsymbol{x} \in \mathbb{R}^m$. Note that $\mathsf{LSE}(\cdot)$ is the standard log-sum-exp function. Also, define the following three vectors in $\mathbb{R}^m_+$:

$$\begin{aligned}
\boldsymbol{y}^1 &= \left(W_k(\mathcal{S}; \theta\boldsymbol{\mu} + (1-\theta)\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})\colon k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]\right) \\
\boldsymbol{y}^2 &= \left(W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})\colon k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]\right) \\
\boldsymbol{y}^3 &= \left(W_k(\mathcal{S}; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})\colon k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]\right)
\end{aligned}$$

By convexity of W_k from part (a), we have that $\boldsymbol{y}^1 \leq \theta\boldsymbol{y}^2 + (1-\theta)\boldsymbol{y}^3$, where the inequality holds componentwise. By definition,

$$\begin{aligned}
\frac{1}{\lambda_j} W_j(\mathcal{S}; \theta\boldsymbol{\mu} + (1-\theta)\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) &= \mathsf{LSE}\left(a\boldsymbol{y}^1\right) \\
&\leq \mathsf{LSE}\left(\theta a\boldsymbol{y}^2 + (1-\theta)a\boldsymbol{y}^3\right) && [\mathsf{LSE} \text{ is strictly increasing}] \\
&\leq \theta\mathsf{LSE}\left(a\boldsymbol{y}^2\right) + (1-\theta)\mathsf{LSE}\left(a\boldsymbol{y}^3\right) && [\mathsf{LSE} \text{ is convex}] \\
&= \frac{\theta}{\lambda_j} W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) + \frac{1-\theta}{\lambda_j} W_j(\mathcal{S}; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) \\
&= \frac{1}{\lambda_j} W_j(\mathcal{S}; \theta\boldsymbol{\mu} + (1-\theta)\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) && [\text{by hypothesis (7)}].
\end{aligned}$$

Because the left and right hand side expressions are equal to each other, it must be true that both inequalities hold with equalities. We have thus shown that

$$\mathsf{LSE}\left(a\boldsymbol{y}^1\right) = \mathsf{LSE}\left(\theta a\boldsymbol{y}^2 + (1-\theta)a\boldsymbol{y}^3\right) = \theta\mathsf{LSE}\left(a\boldsymbol{y}^2\right) + (1-\theta)\mathsf{LSE}\left(a\boldsymbol{y}^3\right)$$

We now derive the implications from the above two equalities. Because the LSE function is strictly increasing, the first equality implies that $\boldsymbol{y}^1 = \theta\boldsymbol{y}^2 + (1-\theta)\boldsymbol{y}^3$. In other words, we have that for each $k \in \mathsf{Children}(j)$,

$$y^1_k = \theta y^2_k + (1-\theta)y^3_k \quad\Longleftrightarrow\quad W_k(\mathcal{S}; \theta\boldsymbol{\mu} + (1-\theta)\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) = \theta W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) + (1-\theta)W_k(\mathcal{S}; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})\ .$$

It now follows from the induction hypothesis, applied to W_k, that for each $k \in \mathsf{Children}(j)$, there exists $\xi_k \in \mathbb{R}$ such that $\bar{\mu}_\ell = \mu_\ell + \xi_k$ for all $\ell \in \mathsf{T}_k[\mathcal{S}] \cap \mathcal{S}$. To establish our result, it is now sufficient to show that $\xi_k = \xi_{k'}$ for all $k \neq k'$ such that $\{k, k'\} \subseteq \mathsf{Children}(j)$. For that, we first note that because $\bar{\boldsymbol{\mu}}_{\mathsf{T}_k[\mathcal{S}]\cap\mathcal{S}} = \xi_k + \boldsymbol{\mu}_{\mathsf{T}_k[\mathcal{S}]\cap\mathcal{S}}$, we have that

$$y^3_k = W_k(\mathcal{S}; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) = W_k(\mathcal{S}; \bar{\boldsymbol{\mu}}_{\mathsf{T}_k\cap\mathcal{S}}, \boldsymbol{\lambda}) = \xi_k + W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) = \xi_k + y^2_k\ , \tag{8}$$

where third equality follows from Lemma B.1. Now, if $\boldsymbol{y}^2 = \boldsymbol{y}^3$, then it follows that $\xi_k = 0$ for all $k \in \mathsf{Children}(j)$, establishing the result. Therefore, we assume that $\boldsymbol{y}^2 \neq \boldsymbol{y}^3$.

Then, we focus on the second equality above:

$$\mathsf{LSE}\left(\theta a\boldsymbol{y}^2 + (1-\theta)a\boldsymbol{y}^3\right) \;=\; \theta\mathsf{LSE}\left(a\boldsymbol{y}^2\right) + (1-\theta)\mathsf{LSE}\left(a\boldsymbol{y}^3\right). \tag{9}$$

It is a well-known result that the the function $t \mapsto \mathsf{LSE}(\boldsymbol{x} + t\boldsymbol{z})$ is strictly convex if and only if $\boldsymbol{z} \neq \boldsymbol{e}$. In other words, for some $\boldsymbol{x}, \boldsymbol{z} \in \mathbb{R}^m$, and for any $t_1, t_2 \in \mathbb{R}$, we have that

$$\mathsf{LSE}(\boldsymbol{x} + (\theta t_1 + (1-\theta)t_2)\boldsymbol{z}) = \theta\mathsf{LSE}(\boldsymbol{x} + t_1\boldsymbol{z}) + (1-\theta)\mathsf{LSE}(\boldsymbol{x} + t_2\boldsymbol{z}) \quad \text{ if and only if } \quad \boldsymbol{z} = \boldsymbol{e}.$$

Now choosing $\boldsymbol{x}$, $\boldsymbol{z}$, t_1, and t_2 to satisfy

$$\boldsymbol{x} + t_1\boldsymbol{z} \;=\; a\boldsymbol{y}^2 \qquad \text{and} \qquad \boldsymbol{x} + t_2\boldsymbol{z} \;=\; a\boldsymbol{y}^3,$$

we obtain from (9) that $\mathsf{LSE}(\boldsymbol{x} + (\theta t_1 + (1-\theta)t_2)\boldsymbol{z}) = \theta\mathsf{LSE}(\boldsymbol{x} + t_1\boldsymbol{z}) + (1-\theta)\mathsf{LSE}(\boldsymbol{x} + t_2\boldsymbol{z})$. Therefore, we must have that $\boldsymbol{z} = \boldsymbol{e}$. Solving for $\boldsymbol{z}$, we get that $\boldsymbol{e} = \boldsymbol{z} = a(\boldsymbol{y}^2 - \boldsymbol{y}^3)/(t_1 - t_2)$; this equality is well defined because $\boldsymbol{y}^2 \neq \boldsymbol{y}^3$ ensures that $t_1 \neq t_2$. As a result, for an appropriately defined constant δ, we obtain that $y_k^2 - y_k^3 = \delta$ for all children $k \in \mathsf{Children}(j)$. Because we also have that $y_2^k - y_3^k = \xi_k$ from (8), it must be that $\xi_k = \delta$ for all $k \in \mathsf{Children}(j)$. Consequently, we have shown that $\bar{\mu}_\ell = \mu_\ell + \delta$ for all leaf nodes $\ell \in \cup_{k \in \mathsf{Children}(j)} \mathsf{T}_k[\mathcal{S}] \cap \mathcal{S}$. Because $\mathsf{T}_j[\mathcal{S}] \cap \mathcal{S} = \cup_{k \in \mathsf{Children}(j)} \mathsf{T}_k[\mathcal{S}] \cap \mathcal{S}$, we have shown that $\bar{\mu}_\ell = \mu_\ell + \xi$ for all leaf nodes $\ell \in \mathsf{T}_j[\mathcal{S}] \cap \mathcal{S}$ and $\xi = \delta$, which is the desired result. This completes the necessity part and finishes the induction. Therefore, part (b) holds for all nodes $j \in \mathsf{T}[\mathcal{S}]$.

Proof of part (c): Part (c) follows immediately from parts (a) and (b). ■

B.3 Proof of Lemma B.3

Proof: We will prove each of the three parts separately by induction on the height of node j.

Proof of part (a): The base case where j has a height of zero is trivially true by definition. Suppose the result is true for all nodes of height at most H. Then, consider node j with height $H+1$. By definition,

$$W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) = \lambda_j \log\left(\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_j}\right)$$

Note that $W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ only depends on $(\mu_\ell : \ell \in \mathcal{S})$, $\left(\boldsymbol{\lambda}_{\mathsf{T}_k[\mathcal{S}]} : k \in \mathsf{Children}(j)\right)$, and λ_j. By induction, for each $k \in \mathsf{Children}(j)$, $W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ is increasing in $\boldsymbol{\mu}$ and $\boldsymbol{\lambda}_{\mathsf{T}_k[\mathcal{S}]}$. Moreover, taking the derivative of the above expression with respect to λ_j, we get

$$\frac{\partial W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})}{\partial \lambda_j} = \log\left(\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_j}\right) - \frac{1}{\lambda_j} \cdot \frac{\sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}} W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \times e^{W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_j}}{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_j}} \;\geq\; 0\,,$$

where the inequality follows because $\log\left(\sum_{i=1} e^{x_i}\right) \geq \max_{i=1,\ldots,n} x_i$. This shows that $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is increasing in $(\boldsymbol{\mu},\boldsymbol{\lambda})$, completing the induction and proving part (a).

Proof of part (b): By using induction on the height on node j, we will establish that $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is strictly increasing in μ_ℓ for all $\ell \in \mathsf{T}_j[\mathcal{S}] \cap \mathcal{S}$. For the base case, suppose that j has a height of zero, so $j = \ell$ for some leaf node $\ell \in \mathsf{T}[\mathcal{S}] \cap \mathcal{S}$. We have by definition that $W_\ell(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \mu_\ell$, which is clearly strictly increasing in μ_ℓ. Because $\mathsf{T}_\ell[\mathcal{S}] \cap \mathcal{S} = \{\ell\}$, we have established the base case. Suppose the result is true for all nodes k with height at most H. Consider a non-leaf node j at height $H+1$. By definition, $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \lambda_j \log\left(\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}\right)$. Now consider a leaf node $\ell \in \mathsf{T}_j[\mathcal{S}] \cap \mathcal{S}$. There exists some child node k of j in $\mathsf{T}_j[\mathcal{S}]$ such that $\ell \in \mathsf{T}_k[\mathcal{S}] \cap \mathcal{S}$. Since the height of $k \in \mathsf{Children}(j)$ is at most H, it follows from the induction hypothesis that $W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is strictly increasing in μ_ℓ. Moreover, because the log-sum-exp function is strictly increasing, it follows from the expression for $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ that it is strictly increasing in μ_ℓ. This completes the induction, establishing the monotonicity in $\boldsymbol{\mu}$ for all nodes $j \in \mathsf{T}[\mathcal{S}]$.

Proof of part (c): Now, consider the last result that $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is strictly increasing in λ_k if node k has at least two children in $\mathsf{T}_j[\mathcal{S}]$. For the base case, consider node j with height one, so its children are leaf nodes. By definition, $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \lambda_j \log\left(\sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}} e^{\mu_\ell/\lambda_j}\right)$. Taking the derivative of the above function with respect to λ_j, we get

$$\frac{\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \lambda_j} \;=\; \log\left(\sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}} e^{\mu_\ell/\lambda_j}\right) \;-\; \frac{1}{\lambda_j} \cdot \frac{\sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}} \mu_\ell \times e^{\mu_\ell/\lambda_j}}{\sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}} e^{\mu_\ell/\lambda_j}}\,. \tag{10}$$

Because $|\mathsf{Children}(j) \cap \mathcal{S}| \geq 2$, we have that $\log(\sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}} e^{\mu_\ell/\lambda_j}) > \max_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}} \mu_\ell/\lambda_j$. Further, since the second term in the expression above is a weighted average of the set of numbers $\{\mu_\ell/\lambda_j : \ell \in \mathsf{Children}(j) \cap \mathcal{S}\}$, it follows that $\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial \lambda_j > 0$, which establishes the base case.

Suppose that the result is true for all nodes k with height at most H. Consider a non-leaf node j at height $H+1$. Consider an arbitrary non-leaf node $i \in \mathsf{T}_j[\mathcal{S}]$ such that i has at least two children. If $i \neq j$, then $i \in \mathsf{T}_k[\mathcal{S}]$ for some child k of j in $\mathsf{T}[\mathcal{S}]$. Because i has at least two children in $\mathsf{T}[\mathcal{S}]$, it follows from the induction hypothesis that $W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ is strictly increasing in λ_i. Moreover, because the log-sum-exp function is strictly increasing, it follows from the expression for $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ that it is also strictly increasing in λ_i.

Now suppose $i = j$. Because $W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ are independent of λ_j for all children k of j in sub-tree $\mathsf{T}[\mathcal{S}]$, we can compute the partial derivative of W_j with respect to λ_j, as done for the base case. Using identical arguments, we can conclude that $\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial \lambda_j > 0$, establishing the induction step. This completes the proof. ■

B.4 Proof of Lemma B.4

Proof: Fix an arbitrary $\mathcal{S} \subseteq \mathcal{N}$. We first consider the derivative with respect to μ_ℓ and prove it using induction on the height of the node. For the base case, we consider a leaf node ℓ of sub-tree $\mathsf{T}[\mathcal{S}]$. We have by definition that $W_\ell(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \mu_\ell$. Therefore, $\partial W_\ell(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\mu_\ell = 1 = \psi_{\ell\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$. We have thus established the base case.

Now suppose the result is true for all nodes of height at most H. In other words, suppose that $\partial W_v(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\mu_\ell = \psi_{v\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ for all nodes v of height at most H and all leaf nodes ℓ in $\mathsf{T}[\mathcal{S}]$. Now consider a non-leaf node j at height $H+1$. We have by definition that

$$\frac{\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\mu_\ell} = \frac{\partial}{\partial\mu_\ell}\lambda_j \log\left(\sum_{k\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}\right) = \frac{\sum_{k\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j} \times \frac{\partial W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\mu_\ell}}{\sum_{k\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}}$$

Since ℓ is a leaf node in $\mathsf{T}_j[\mathcal{S}]$, there exists exactly one child node $i \in \mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]$ such that ℓ is a leaf node in $\mathsf{T}_i[\mathcal{S}]$. For any other child node $k \neq i$, we must have that $\partial W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\mu_\ell = 0$. Therefore, we obtain

$$\frac{\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\mu_\ell} = \frac{e^{W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}}{\sum_{k\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}} \times \frac{\partial W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\mu_\ell} = \psi_{j\to i}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \psi_{i\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}),$$

where the first term in the second equality above follows from the definition of $\psi_{j\to i}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ and the second term follows from the induction hypothesis. Because $\psi_{j\to i}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \psi_{i\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \psi_{j\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ by definition, we have shown that $\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\mu_\ell = \psi_{j\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$, as desired. We have thus established the induction step. The first result now follows by induction.

We now consider the derivative of $W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ with respect to λ_k for some non-leaf nodes j and k in $\mathsf{T}[\mathcal{S}]$. We prove the result by induction on the height of j. For the base case, we start with a non-leaf node j at height one, so all children of j are leaf nodes. As shown in Equation (10) in the proof of Lemma B.3, we have that

$$\frac{\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_j} = \log\left(\sum_{\ell\in\mathsf{Children}(j)\cap\mathcal{S}} e^{\mu_\ell/\lambda_j}\right) - \frac{1}{\lambda_j}\cdot\frac{\sum_{\ell\in\mathsf{Children}(j)\cap\mathcal{S}} \mu_\ell \times e^{\mu_\ell/\lambda_j}}{\sum_{\ell\in\mathsf{Children}(j)\cap\mathcal{S}} e^{\mu_\ell/\lambda_j}} = \psi_{j\to j}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \Delta_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}),$$

where the second equality follows from the definition of $\Delta_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ and the fact that $\psi_{j\to j}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = 1$. We have thus established the base case.

For the induction step, we suppose that $\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\lambda_k = \psi_{j\to k}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \Delta_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ for all nodes j of height at most H and for all non-leaf nodes $k \in \mathsf{T}_j[\mathcal{S}]$. Now consider a node j of height $H+1$. First, suppose that $k \neq j$. We then have by definition that

$$\frac{\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k} = \frac{\partial}{\partial\lambda_k}\lambda_j \log\left(\sum_{i\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}\right) = \frac{\sum_{i\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j} \times \frac{\partial W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k}}{\sum_{i\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}}.$$

Because of the property of a tree, the non-leaf node k belongs to the sub-tree $\mathsf{T}_i[\mathcal{S}]$ of exactly one child node i of node j. For any other child node $v \neq i$, we have that $\partial W_v(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\lambda_k = 0$. Therefore, we can now write

$$\frac{\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k} = \frac{e^{W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}}{\sum_{i\in\,\mathsf{Children}(j)\,\cap\,\mathsf{T}[\mathcal{S}]} e^{W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\lambda_j}} \times \frac{\partial W_i(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k} = \psi_{j\to i}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \psi_{i\to k}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \Delta_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}),$$

where the first term in the second equality above follows from the definition on $\psi_{j\to i}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ and the second and third terms follow from the induction hypothesis. Because $\psi_{j\to k}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \psi_{j\to i}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \psi_{i\to k}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ by definition, we have shown that $\partial W_j(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\lambda_k = \psi_{j\to k}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) \times \Delta_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$, as desired. This completes the induction step, completing the lemma. The final expression for $\Delta_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ follows from plugging in the definitions of $W_k(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ and $\psi_{k\to i}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$ for all $i \in \mathsf{Children}(k) \cap \mathsf{T}[\mathcal{S}]$. ■

It follows from the above lemma that $\frac{\partial W_{\mathsf{root}}}{\partial\mu_\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \psi_{\mathsf{root}\to\ell}(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda}) = \mathbb{P}_\ell(\mathcal{S};\boldsymbol{\mu},\boldsymbol{\lambda})$, and this is consistent with the result from McFadden (1978), which provides an expression of the choice probability in terms of the derivative of the generating function for Generalized Extreme Value choice models.

Appendix C: Derivatives of the Negative log-likelihood function

The following lemmas show that the partial derivatives $(\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\mu_\ell : \ell \in \mathcal{N})$ and $(\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\lambda_j : j \in \mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\}))$ can be computed efficiently via a recursion that starts at the root node and ends at the leaf nodes of the tree T. To facilitate our exposition, for each transaction q, define the set of values $\left(D_j^q(\boldsymbol{\mu},\boldsymbol{\lambda}) : j \in \mathsf{T}\right)$ recursively starting from root as follows: Initialize $D^q_{\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}) = 1$ and for all $j \in \mathsf{T}\setminus\{\mathsf{root}\}$,

$$D_j^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \equiv \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) + \psi^q_{\mathsf{pa}(j)\to j}(\boldsymbol{\mu},\boldsymbol{\lambda}) \cdot D^q_{\mathsf{pa}(j)}(\boldsymbol{\mu},\boldsymbol{\lambda}) \ .$$

We emphasize that the values $D_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})$ can be computed efficiently through a simple recursion starting from the root node. We will use the values $D_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})$ to compute the derivative of the log-likeihood function; see Lemmas C.2 and C.3. The following lemma gives an equivalent expression for $D_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})$.

Lemma C.1 (Equivalent Expression for $D_j^q(\boldsymbol{\mu},\boldsymbol{\lambda})$) *For each transaction q and for all $k \in \mathsf{T}$, $D_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) = \sum_{j\in\mathsf{T}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \cdot \psi^q_{j\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})$.*

Proof: We prove the result by induction on the height of node k. When $k = \mathsf{root}$, the RHS in the equality above reduces to

$$\begin{aligned}\sum_{j\in\mathsf{T}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \cdot \psi^q_{j\to\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}) &= \left(\frac{1}{\lambda_{\mathsf{root}}} - \frac{1}{\lambda_{\mathsf{pa(root)}}}\right) \cdot \psi^q_{\mathsf{root}\to\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}) \\ &= \psi^q_{\mathsf{root}\to\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}) \ = \ 1 \ = \ D^q_{\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}),\end{aligned}$$

where the first equality follows since $\psi^q_{j\to\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda})=0$ for all $j\neq\mathsf{root}$, and the second follows since $\lambda_{\mathsf{pa(root)}}=+\infty$. This establishes the base case. Now, suppose the claim is true for all nodes k of height H. Consider any node k with height $H-1$. Using the definition of $D^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})$, it follows that

$$\begin{aligned}
D^q_k(\boldsymbol{\mu},\boldsymbol{\lambda}) &= \mathbb{1}_{\{c^q\in\mathsf{T}_k\}}\cdot\left(\frac{1}{\lambda_k}-\frac{1}{\lambda_{\mathsf{pa}(k)}}\right)+\psi^q_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\cdot D^q_{\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})\\
&= \mathbb{1}_{\{c^q\in\mathsf{T}_k\}}\cdot\left(\frac{1}{\lambda_k}-\frac{1}{\lambda_{\mathsf{pa}(k)}}\right)+\psi^q_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\cdot\left(\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\cdot\psi^q_{j\to\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})\right)\\
&= \mathbb{1}_{\{c^q\in\mathsf{T}_k\}}\cdot\left(\frac{1}{\lambda_k}-\frac{1}{\lambda_{\mathsf{pa}(k)}}\right)+\psi^q_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\cdot\left(\sum_{j\in\mathsf{T}\setminus\{k\}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\cdot\psi^q_{j\to\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})\right)\\
&= \mathbb{1}_{\{c^q\in\mathsf{T}_k\}}\cdot\left(\frac{1}{\lambda_k}-\frac{1}{\lambda_{\mathsf{pa}(k)}}\right)+\sum_{j\in\mathsf{T}\setminus\{k\}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\cdot\psi^q_{j\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\\
&= \sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\cdot\psi^q_{j\to k}(\boldsymbol{\mu},\boldsymbol{\lambda}),
\end{aligned}$$

where the second equality follows from the induction hypothesis, the third since $\psi^q_{k\to\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})=0$, the fourth since $\psi^q_{j\to\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda})\times\psi^q_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})=\psi^q_{j\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})$ for all $j\in\mathsf{path}[\mathsf{root},k)$, and the final since $\psi^q_{k\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})=1$. This completes the induction. ■

Here is the derivative of the negative log-likelihood function with respect to $\boldsymbol{\mu}$.

Lemma C.2 (Derivatives with Respect to $\boldsymbol{\mu}$) *For all $\ell\in\mathcal{N}$,*

$$\frac{\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\mu_\ell} \;=\; \frac{1}{Q}\sum_{q=1}^{Q}D^q_\ell(\boldsymbol{\mu},\boldsymbol{\lambda}) \;=\; \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\mathcal{N}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\cdot\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\cdot\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})-\frac{\mathsf{sales}_\ell}{\lambda_{\mathsf{pa}(\ell)}}$$

Proof. From Theorem 2.2, it follows that

$$\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})=\frac{1}{Q}\sum_{q=1}^{Q}\left\{\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})\right\}.$$

Now, noting that $\partial W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})/\partial\mu_\ell$ is equal to $\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})$ (from Lemma B.4 in Appendix B), it follows that for all $\ell\in\mathcal{N}$:

$$\frac{\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\mu_\ell}=\frac{1}{Q}\sum_{q=1}^{Q}\left\{\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})\right\}=\frac{1}{Q}\sum_{q=1}^{Q}D^q_\ell(\boldsymbol{\mu},\boldsymbol{\lambda}),$$

where the second equality follows from Lemma C.1. Finally, we note that

$$\begin{aligned}
\frac{1}{Q}\sum_{q=1}^{Q}D^q_\ell(\boldsymbol{\mu},\boldsymbol{\lambda}) &= \frac{1}{Q}\sum_{q=1}^{Q}\left\{\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})\right\}\\
&= \frac{1}{Q}\sum_{q=1}^{Q}\left\{\sum_{j\in\mathsf{T}\setminus\mathcal{N}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})+\sum_{j\in\mathcal{N}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})\right\}
\end{aligned}$$

$$
\begin{aligned}
&= \frac{1}{Q}\sum_{q=1}^{Q}\left\{\sum_{j\in\mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) - \frac{\mathbb{1}_{\{c^q=\ell\}}}{\lambda_{\mathsf{pa}(\ell)}}\right\}\\
&= \frac{1}{Q}\sum_{q=1}^{Q}\left\{\sum_{j\in\mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})\right\} - \frac{1}{Q}\sum_{q=1}^{Q}\frac{\mathbb{1}_{\{c^q=\ell\}}}{\lambda_{\mathsf{pa}(\ell)}}\\
&= \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\cdot\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\cdot\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) - \frac{\mathsf{sales}_\ell}{\lambda_{\mathsf{pa}(\ell)}},
\end{aligned}
$$

where the third equality follows since $\lambda_j = +\infty$ for all $j\in\mathcal{N}$ and $\psi^q_{j\to\ell}=0$ for all $j\in\mathcal{N}; j\neq\ell$, and the last follows from the definition of sales_ℓ. ■

The next lemma gives the derivative with respect to $\boldsymbol{\lambda}$.

Lemma C.3 (Derivatives with Respect to $\boldsymbol{\lambda}$) *For all $k\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})$,*

$$
\frac{\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k} = \frac{1}{Q}\sum_{q=1}^{Q}\Delta^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})\cdot D^q_k(\boldsymbol{\mu},\boldsymbol{\lambda}) + E^q_k(\boldsymbol{\mu},\boldsymbol{\lambda}),
$$

where for all q, $\Delta^q_k(\boldsymbol{\mu},\boldsymbol{\lambda}) := \Delta_k(S^q;\boldsymbol{\mu},\boldsymbol{\lambda})$ is as defined in Lemma B.4, and

$$
E^q_k(\boldsymbol{\mu},\boldsymbol{\lambda}) = \frac{\sum_{i\in\,\mathsf{Children}(k)}\mathbb{1}_{\{c^q\in\mathsf{T}_i\}}\log\left(\psi^q_{k\to i}(\boldsymbol{\mu},\boldsymbol{\lambda})\right)}{\lambda_k}.
$$

Proof. From Theorem 2.2, it follows that

$$
\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda}) = \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_j} \;-\; \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}}
$$

From the quotient rule of derivatives, it follows that

$$
\begin{aligned}
\frac{\partial}{\partial\lambda_k}\left(\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_j}\right) &= \frac{1}{\lambda_j}\cdot\frac{\partial W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k} - \mathbb{1}_{\{j=k\}}\frac{W^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_k^2}\\
\frac{\partial}{\partial\lambda_k}\left(\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}}\right) &= \frac{1}{\lambda_{\mathsf{pa}(j)}}\cdot\frac{\partial W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k} - \mathbb{1}_{\{\mathsf{pa}(j)=k\}}\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_k^2}
\end{aligned}
$$

Using the above and the expression for $\frac{\partial W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k}$ from Lemma B.4, it follows that

$$
\begin{aligned}
\frac{\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_k} &= \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\cdot\left(\frac{\psi^q_{j\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\Delta^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_j} - \mathbb{1}_{\{k=j\}}\frac{W^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_k^2}\right)\\
&\quad - \frac{1}{Q}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\cdot\left(\frac{\psi^q_{j\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\Delta^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}} - \mathbb{1}_{\{k=\mathsf{pa}(j)\}}\frac{W^q_j(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_k^2}\right)\\
&= \frac{1}{Q}\sum_{q=1}^{Q}\Delta^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})\cdot\left(\sum_{j\in\mathsf{T}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\frac{1}{\lambda_j}-\frac{1}{\lambda_{\mathsf{pa}(j)}}\right)\cdot\psi^q_{j\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})\right)\\
&\quad - \frac{1}{Q}\sum_{q=1}^{Q}\left(\mathbb{1}_{\{c^q\in\mathsf{T}_k\}}W^q_k(\boldsymbol{\mu},\boldsymbol{\lambda}) - \sum_{i\in\,\mathsf{Children}(k)}\mathbb{1}_{\{c^q\in\mathsf{T}_i\}}W^q_i(\boldsymbol{\mu},\boldsymbol{\lambda})\right)\frac{1}{\lambda_k^2}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{Q}\sum_{q=1}^{Q} \Delta_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \cdot D_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \;-\; \frac{1}{Q}\sum_{q=1}^{Q} \sum_{i\in \mathsf{Children}(k)} \mathbb{1}_{\{c^q\in \mathsf{T}_i\}} \cdot (W_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) - W_i^q(\boldsymbol{\mu},\boldsymbol{\lambda})) \cdot \frac{1}{\lambda_k^2} \\
&= \frac{1}{Q}\sum_{q=1}^{Q} \Delta_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \cdot D_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \;-\; \frac{1}{Q}\sum_{q=1}^{Q} \sum_{i\in \mathsf{Children}(k)} \mathbb{1}_{\{c^q\in \mathsf{T}_i\}} \frac{-\log(\psi_{k\to i}^q(\boldsymbol{\mu},\boldsymbol{\lambda}))\cdot\lambda_k}{\lambda_k^2} \\
&= \frac{1}{Q}\sum_{q=1}^{Q} \Delta_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \cdot D_k^q(\boldsymbol{\mu},\boldsymbol{\lambda}) \;+\; \frac{1}{Q}\sum_{q=1}^{Q} E_k^q(\boldsymbol{\mu},\boldsymbol{\lambda})
\end{aligned}
$$

where the third equality follows from the expression for $D_k^q(\boldsymbol{\mu},\boldsymbol{\lambda})$ in Lemma C.1 and the fact that $\mathbb{1}_{\{c^q\in \mathsf{T}_k\}} = \sum_{i\in \mathsf{Children}(k)} \mathbb{1}_{\{c^q\in \mathsf{T}_i\}}$ for all q, the fourth equality follows from the definition of $\psi_{k\to i}^q(\boldsymbol{\mu},\boldsymbol{\lambda})$ and the last follows from the definition of $E_k^q(\boldsymbol{\mu},\boldsymbol{\lambda})$ in the statement of the lemma.

Appendix D: Proofs of Theorem 3.2 and Theorem 3.3

D.1 Proof of Theorem 3.2

To facilitate the proof of Theorem 3.2, let us introduce the following function $b : \mathcal{D}_1 \to \mathbb{R}_+$ defined by: for each $\boldsymbol{\zeta} \in \mathcal{D}_1$,

$$
b(\boldsymbol{\zeta}) = \max_{q=1,\ldots,Q} \; \max_{\ell\in\mathcal{S}^q\setminus\{c^q\}} \; (\zeta_\ell - \zeta_{c^q}) \;,
$$

where we use the convention that the maximum of an empty set is 0. Further, we let $b^* = \min_{\boldsymbol{\zeta}\in\mathcal{D}_1\cap\Delta} b(\boldsymbol{\zeta})$ where $\Delta = \{\boldsymbol{\mu} \mid \|\boldsymbol{\mu}\|_\infty = 1\}$. In other words, $b(\boldsymbol{\zeta})$ denotes the maximum possible loss in mean utility from the choices made in the data under the mean utility vector $\boldsymbol{\zeta}$. Maximizing the log-likelihood requires us to make this loss negative, *if* possible. If the loss is made negative for some $\boldsymbol{\zeta}$, then scaling all the mean utility values by a contant makes the loss diverge to $-\infty$, resulting in unbounded optimal solutions. As we shown below, the assumption of strong connectedness of the comparison graph prevents this from happening. In particular, strong connectedness ensures that the loss is always positive for any $\boldsymbol{\zeta}$. We formalize this intuition next. The first lemma shows that b^* is positive when the comparison graph Comp is strongly connected. Recall that, given transaction data $\{(S^q, c^q) : q = 1,\ldots,Q\}$, a comparison graph $\mathsf{Comp} = (\mathcal{N}, \mathsf{E})$ is a directed graph whose nodes correspond to the products, and there is a directed edge $(\ell_1,\ell_2) \in \mathsf{E}$ if there exists an offer set $\mathcal{S}^q$ such that $\{\ell_1,\ell_2\} \subseteq \mathcal{S}^q$ and $c^q = \ell_1$.

Lemma D.1 (Positive Gap) *If the comparison graph* Comp *is strongly connected, then* $b^* > 0$.

Proof: We start with the observation that because $\Delta \cap \mathcal{D}_1$ is compact, the minimum $\min_{\boldsymbol{\mu}\in\Delta\cap\mathcal{D}_1} b(\boldsymbol{\mu})$ is attained at some $\boldsymbol{\mu}^* \in \Delta\cap\mathcal{D}_1$, so that $b^* = b(\boldsymbol{\mu}^*)$. Therefore, to show that $b^* > 0$, it is sufficient to show that $b(\boldsymbol{\mu}) > 0$ for every $\boldsymbol{\mu} \in \Delta\cap\mathcal{D}_1$. We prove this result by contradiction. Suppose on the contrary that $b(\boldsymbol{\zeta}) \le 0$ for some $\boldsymbol{\zeta} \in \Delta\cap\mathcal{D}_1$. Because $b(\boldsymbol{\zeta}) \le 0$, it follows from definition that $\zeta_\ell \le \zeta_{c^q}$ for all $\ell \in \mathcal{S}_q$ and for all $q = 1\ldots,Q$. Consider an arbitrary directed edge

(j,k) in Comp. By our construction, there exists a $q \in \{1,\ldots,Q\}$ such that $j = c^q$ and $k \in \mathcal{S}^q \setminus \{c^q\}$. Therefore, we must have that $\zeta_k \le \zeta_j$. But, Comp is strongly connected, which implies that there is a directed path $k \to s_1 \to s_2 \to \cdots s_m \to j$ in Comp from node k to j. Applying the result that $\zeta_t \le \zeta_s$ whenever there is a directed edge from s to t, we obtain that $\zeta_j \le \zeta_{s_m} \le \cdots \le \zeta_{s_1} \le \zeta_k$, which implies that $\zeta_j \le \zeta_k$. This inequality together with the result that $\zeta_k \le \zeta_j$, implies that $\zeta_k = \zeta_j$. Because j and k are arbitrary nodes and any two nodes are connected by directed paths, we have shown that $\zeta_1 = \zeta_2 = \cdots = \zeta_n$. But $\boldsymbol{\zeta} \in \Delta \cap \mathcal{D}_1$, so $\zeta_1 = 0$, and consequently that $\mathbf{0} \in \Delta$. This is a contradiction! Thus, $b(\boldsymbol{\mu}) > 0$ for all $\boldsymbol{\mu} \in \Delta \cap \mathcal{D}_1$. This completes the proof. ■

The next lemma gives an upper bound on the selection probability in terms of the mean utilities.

Lemma D.2 *For each $\ell_1 \in \mathcal{S}$, $\ell_2 \in \mathcal{S}$, and $(\boldsymbol{\mu}, \boldsymbol{\lambda})$, $-\log \mathbb{P}_{\ell_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \ge \mu_{\ell_2} - \mu_{\ell_1}$.*

Proof: By Equation (2),

$$
\begin{aligned}
-\log \mathbb{P}_{\ell_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;&=\; -\log \psi_{\mathsf{root} \to \ell_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;=\; \sum_{j \in \mathsf{path}(\mathsf{root}, \ell_1]} \frac{W_{\mathsf{pa}(j)}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) - W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}} \\
&\ge \sum_{j \in \mathsf{path}(\mathsf{root}, \ell_1]} W_{\mathsf{pa}(j)}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) - W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;=\; W_{\mathsf{root}}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) - W_{\ell_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;,
\end{aligned}
$$

where the inequality follows because $W_{\mathsf{pa}(j)}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \ge W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ and $\lambda_{\mathsf{pa}(j)} \le 1$ for each node j such that $j \ne \mathsf{root}$. The last equality follows from the telescoping sum.

Now, we claim that $W_{\mathsf{root}}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \ge W_\ell(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ for any leaf node $\ell \in \mathcal{S}$. To see this, consider the path $\mathsf{root} \to j_1 \to \cdots \to j_m \to \ell$ from the root node to the leaf node ℓ in tree $\mathsf{T}[\mathcal{S}]$. Because $W_{\mathsf{pa}(j)}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \ge W_j(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$ for all nodes j, we have the sequence of inequalities $W_{\mathsf{root}}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \ge W_{j_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \ge \cdots \ge W_{j_m}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \ge W_\ell(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})$, which establishes the claim.

By choosing $\ell = \ell_2$, we get

$$
-\log \mathbb{P}_{\ell_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;\ge\; W_{\mathsf{root}}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) - W_{\ell_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;\ge\; W_{\ell_2}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) - W_{\ell_1}(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;=\; \mu_{\ell_2} - \mu_{\ell_1} \;,
$$

where the last equality follows because both ℓ_1 and ℓ_2 are in $\mathcal{S}$. ■

The next lemma establishes an upper bound on the $\mathsf{NegLog}(\mathbf{0}, \boldsymbol{\lambda})$ for all $\boldsymbol{\lambda}$.

Lemma D.3 *For each $\boldsymbol{\lambda} \in \mathcal{D}_2$, $\mathsf{NegLog}(\mathbf{0}, \boldsymbol{\lambda}) \le |\mathsf{T}| \log |\mathsf{T}|$.*

Proof: Consider an arbitrary $\boldsymbol{\lambda} \in \mathcal{D}_2$ and $\mathcal{S} \subseteq \mathcal{N}$. We will first establish the following claim.

Claim: For each node $j \in \mathsf{T}[\mathcal{S}]$, $W_j(\mathcal{S}; \mathbf{0}, \boldsymbol{\lambda})/\lambda_j \le \log |\mathsf{T}_j[\mathcal{S}]|$.

We will prove the claim by induction on the height of node j. For the base case where node j has a height of zero, then $j = \ell$ for some leaf node $\ell \in \mathcal{S}$. Since $\lambda_\ell = +\infty$ by definition, we have $W_\ell(\mathcal{S}; \mathbf{0}, \boldsymbol{\lambda})/\lambda_\ell = 0 = \log |\mathsf{T}_\ell[\mathcal{S}]|$ because $\mathsf{T}_\ell[\mathcal{S}] = \{\ell\}$. This establishes the base case. Suppose the

claim holds for all nodes with height at most H. Consider an arbitrary node j with height $H+1$. By the inductive hypothesis, the claim holds for all children of j; that is, for each $k \in \mathsf{Children}(j) \cap \mathsf{T}[\mathcal{S}]$, $W_k(\mathcal{S};\mathbf{0},\boldsymbol{\lambda})/\lambda_k \leq \log|\mathsf{T}_k[\mathcal{S}]|$, and thus, $W_k(\mathcal{S};\mathbf{0},\boldsymbol{\lambda})/\lambda_j \leq \log|\mathsf{T}_k[\mathcal{S}]|$ because $\lambda_k \leq \lambda_j$. Therefore,

$$\frac{W_j(\mathcal{S};\mathbf{0},\boldsymbol{\lambda})}{\lambda_j} = \log\left(\sum_{k\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} e^{W_k(\mathcal{S};\mathbf{0},\boldsymbol{\lambda})/\lambda_j}\right) \leq \log\left(\sum_{k\in\mathsf{Children}(j)\cap\mathsf{T}[\mathcal{S}]} |\mathsf{T}_k[\mathcal{S}]|\right) \leq \log|\mathsf{T}_j[\mathcal{S}]| ,$$

which completes the induction argument. So, the claim holds for all nodes $j \in \mathsf{T}[\mathcal{S}]$.

To establish the lemma, note that by Equation (2), for each transaction $(\mathcal{S}^q, c^q)$,

$$\begin{aligned} -\log\mathbb{P}_{c^q}(\mathcal{S}^q;\mathbf{0},\boldsymbol{\lambda}) &= -\log\psi_{\mathsf{root}\to c^q}(\mathcal{S}^q;\mathbf{0},\boldsymbol{\lambda}) = \sum_{j\in\mathsf{path}(\mathsf{root},c^q]} \frac{W_{\mathsf{pa}(j)}(\mathcal{S}^q;\mathbf{0},\boldsymbol{\lambda}) - W_j(\mathcal{S}^q;\mathbf{0},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}} \\ &\leq \sum_{j\in\mathsf{path}(\mathsf{root},c^q]} \frac{W_{\mathsf{pa}(j)}(\mathcal{S}^q;\mathbf{0},\boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}} \leq \sum_{j\in\mathsf{path}(\mathsf{root},c^q]} \log\left|\mathsf{T}^q_{\mathsf{pa}(j)}\right| \leq |\mathsf{T}^q|\log|\mathsf{T}^q| , \end{aligned}$$

where the first inequality follows because $W_j(\cdot)$ is non-negative, and the second inequality follows from the above claim. Therefore, $\mathsf{NegLog}(\mathbf{0},\boldsymbol{\lambda}) \leq \frac{1}{Q}\sum_{q=1}^{Q}|\mathsf{T}^q|\log|\mathsf{T}^q| \leq |\mathsf{T}|\log|\mathsf{T}|$, which is the desired result. ■

Here is the proof of Theorem 3.2.

Proof of Theorem 3.2: Fix an arbitrary $\boldsymbol{\lambda} \in \mathcal{D}_2$. Recall that $\mathcal{D}_1 = \{\boldsymbol{\mu} \in \mathbb{R}^n : \mu_1 = 0\}$.

Proof of Necessity: We will first prove the necessity, so suppose that the optimization problem $\min_{\boldsymbol{\mu}\in\mathcal{D}_1}\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})$ admits a unique and bounded optimal solution, and let $\boldsymbol{\mu}^* \in \mathcal{D}_1$ denote the unique optimal solution. We will prove by contradiction that the comparison graph Comp must be strongly connected.

Suppose on the contrary that Comp is not strongly connected. This means that there exist vertices ℓ_1 and ℓ_2 such that ℓ_2 is not reachable from ℓ_1. Let A_1 denote the set of vertices that are reachable from ℓ_1, and let $A_2 = \mathcal{N} \setminus A_1$ denote the remaining vertices in Comp. Note that $\ell_1 \in A_1$ and $\ell_2 \in A_2$, so both A_1 and A_2 are nonempty, disjoint, and $A_1 \cup A_2 = \mathcal{N}$. By definition, there is no directed path from a node in A_1 to a node in A_2. Without loss of generality, assume that $1 \in A_1$; the argument for the case where $1 \in A_2$ is analogous.

Consider an arbitrary $\xi > 0$. Define a $\bar{\boldsymbol{\mu}}$ as follows:

$$\bar{\mu}_\ell = \begin{cases} \mu^*_\ell & \text{if} \quad \ell \in A_1, \\ \mu^*_\ell + \xi & \text{if} \quad \ell \in A_2, \end{cases}$$

It is easy to verify that $\bar{\boldsymbol{\mu}} \in \mathcal{D}_1$. Now, consider an arbitrary transaction $(\mathcal{S}^q, c^q)$. There are 3 cases to consider: 1) $\mathcal{S}^q \subseteq A_1$, 2) $\mathcal{S}^q \subseteq A_2$, and 3) $\mathcal{S}^q \cap A_1 \neq \varnothing$ and $\mathcal{S}^q \cap A_2 \neq \varnothing$.

Case 1: $\mathcal{S}^q \subseteq A_1$. Then, we have that $W_j(\mathcal{S}^q;\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}) = W_j(\mathcal{S}^q;\boldsymbol{\mu}^*,\boldsymbol{\lambda})$ for all $j \in \mathsf{T}[\mathcal{S}^q]$ because the weight function only depends on the utility of the products in the offer

set $\mathcal{S}^q$, and by our construction, we have that $\bar{\mu}_\ell = \mu^*_\ell$ for all $\ell \in \mathcal{S}^q \subseteq A_1$. Therefore, $-\log \psi_{\mathsf{root}\to c^q}(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) = -\log \psi_{\mathsf{root}\to c^q}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda})$.

Case 2: $\mathcal{S}^q \subseteq A_2$. By Equation (2), we have that

$$\begin{aligned}
-\log \psi_{\mathsf{root}\to c^q}(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) &= \sum_{j \in \mathsf{path}(\mathsf{root}, c^q]} \frac{W^q_{\mathsf{pa}(j)}(\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) - W^q_j(\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}} \\
&\overset{\text{(a)}}{=} \frac{W_{\mathsf{root}}(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q]} W_j(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) , \\
&\overset{\text{(b)}}{=} \frac{W_{\mathsf{root}}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda})}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q]} W_j(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}) \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \\
&\qquad + \frac{\xi}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q]} \left(\frac{\xi}{\lambda_j} - \frac{\xi}{\lambda_{\mathsf{pa}(j)}}\right) \\
&\overset{\text{(c)}}{=} \frac{W_{\mathsf{root}}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda})}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q]} W_j(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}) \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(v)}}\right) \\
&= -\log \psi_{\mathsf{root}\to c^q}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}) \ .
\end{aligned}$$

where the equality (a) follows from collecting terms and (b) follows because because $\bar{\mu}_\ell = \mu^*_\ell + \xi$ for all $\ell \in A_2$ and from Lemma B.1 on the invariance under translation by a constant, which implies that $W_j(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) = \xi + W_j(\mathcal{S}^q; \boldsymbol{\mu}, \boldsymbol{\lambda})$. The last equality (c) follows by the telescoping sum and the fact that $\lambda_\ell = +\infty$ for all leaf nodes $\ell \in \mathcal{N}$.

Case 3: $\mathcal{S}^q \cap A_1 \neq \varnothing$ and $\mathcal{S}^q \cap A_2 \neq \varnothing$. In this case, $|\mathcal{S}^q| \geq 2$, and thus, by our construction of the comparison graph Comp, there is a directed edge from c^q to all other elements in $\mathcal{S}^q$. Since there is no directed path from A_1 to A_2, it must be the case that $c^q \in A_2$. By re-arranging the terms in Equation (2) and using the fact that $\lambda_\ell = +\infty$ for all leaf nodes $\ell \in \mathcal{N}$, we can write

$$\begin{aligned}
-\log \psi_{\mathsf{root}\to c^q}(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) &= \frac{W_{\mathsf{root}}(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q)} W_j(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) - \frac{W_{c^q}(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(c^q)}} \\
&\overset{\text{(a)}}{=} \frac{W_{\mathsf{root}}(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda})}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q)} W_j(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \\
&\qquad - \frac{W_{c^q}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(c^q)}} - \frac{\xi}{\lambda_{\mathsf{pa}(c^q)}} \\
&\overset{\text{(b)}}{\leq} \frac{W_{\mathsf{root}}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda})}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q)} W_j(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}) \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \\
&\qquad + \frac{\xi}{\lambda_{\mathsf{root}}} + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q)} \left(\frac{\xi}{\lambda_j} - \frac{\xi}{\lambda_{\mathsf{pa}(j)}}\right) - \frac{W_{c^q}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(c^q)}} - \frac{\xi}{\lambda_{\mathsf{pa}(c^q)}} \\
&\overset{\text{(c)}}{=} -\log \psi_{\mathsf{root}\to c^q}(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}) \ .
\end{aligned}$$

where (a) follows because $c_q \in A_2$ for all q, so $W_{c^q}\left(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}\right) = \bar{\mu}_{c^q} = \mu^*_{c^q} + \xi$. The inequality (b) above follows from the fact for all $j \in \mathsf{path}(\mathsf{root}, c^q)$,

$$W_j\left(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}\right)\left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \;\leq\; W_j\left(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}\right)\left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \;+\; \xi\left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right)$$

because $0 < \lambda_j \leq \lambda_{\mathsf{pa}(j)}$ and by Lemma B.3, the weight function is increasing in $\boldsymbol{\mu}$, so for all $j \in \mathsf{T}[\mathcal{S}^q]$, $W_j\left(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}\right) \;\leq\; W_j\left(\mathcal{S}^q; \boldsymbol{\mu}^* + \xi \boldsymbol{e}, \boldsymbol{\lambda}\right) \;=\; \xi + W_j\left(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}\right)$, where the equality follows from the translation variance property. The final equality (c) above follows from collecting terms in the telescoping sum.

Therefore, we observe that in all three cases,

$$-\log \psi_{\mathsf{root}\to c^q}\left(\mathcal{S}^q; \bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}\right) \;\leq\; -\log \psi_{\mathsf{root}\to c^q}\left(\mathcal{S}^q; \boldsymbol{\mu}^*, \boldsymbol{\lambda}\right) \;.$$

Since the transaction $(\mathcal{S}^q, c^q)$ is arbitrary, it follows that $\mathsf{NegLog}(\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}) \leq \mathsf{NegLog}(\boldsymbol{\mu}^*, \boldsymbol{\lambda})$, but this contradicts the fact that $\boldsymbol{\mu}^*$ is the unique optimal solution! Hence, it must be the case that Comp is strongly connected. This completes the proof of the necessity.

Proof of Sufficiency: We will now prove the sufficiency, so assume that the comparison graph Comp is strongly connected. Fix an arbitrary $\boldsymbol{\lambda}$. Consider $\boldsymbol{\mu} \in \mathcal{D}_1$ such that $\|\boldsymbol{\mu}\|_\infty > \frac{Q}{b^*}\mathsf{NegLog}(\mathbf{0}, \boldsymbol{\lambda})$. Note that $b^* > 0$ by Lemma D.1. Then, $\boldsymbol{\mu} = \|\boldsymbol{\mu}\|_\infty \boldsymbol{\zeta}$ where $\|\boldsymbol{\zeta}\|_\infty = 1$. By definition, there exists a transaction $(\mathcal{S}^q, c^q)$ such that $b(\boldsymbol{\zeta}) = \zeta_\ell \;-\; \zeta_{c^q}$ for some $\ell \in \mathcal{S}^q \setminus \{c^q\}$. Then, by Lemma D.2

$$\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda}) \;\geq\; -\log \mathbb{P}_{c^q}\left(\mathcal{S}^q; \boldsymbol{\mu}, \boldsymbol{\lambda}\right)/Q \;\geq\; \left(\mu_\ell - \mu_{c^q}\right)/Q \;=\; \left(\zeta_\ell - \zeta_{c^q}\right)\|\boldsymbol{\mu}\|_\infty/Q \;=\; b(\boldsymbol{\zeta})\,\|\boldsymbol{\mu}\|_\infty/Q,$$

and it follows that $\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda}) \;\geq\; \|\boldsymbol{\mu}\|_\infty b(\boldsymbol{\zeta})/Q \;\geq\; \|\boldsymbol{\mu}\|_\infty b^*/Q \;>\; \mathsf{NegLog}(\mathbf{0}, \boldsymbol{\lambda})$, which shows that $\min_{\boldsymbol{\mu}\in\mathcal{D}_1} \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \min\left\{\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda}) \,\middle|\, \boldsymbol{\mu} \in \mathcal{D}_1,\; \|\boldsymbol{\mu}\|_\infty \leq \frac{Q}{b^*}\mathsf{NegLog}(\mathbf{0}, \boldsymbol{\lambda})\right\}$, so the optimization problem has a bounded solution.

We now show that the optimal solution must be unique by establishing that the mapping $\boldsymbol{\mu} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda})$ is strictly convex over the set $\mathcal{D}_1$. For that, we first note from Theorem 2.2 that

$$\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda}) = \frac{1}{Q}\sum_{q=1}^{Q}\left\{W^q_{\mathsf{root}}\left(\boldsymbol{\mu}, \boldsymbol{\lambda}\right) \;+\; \sum_{j\in\mathsf{path}(\mathsf{root}, c^q)} W^q_j\left(\boldsymbol{\mu}, \boldsymbol{\lambda}\right)\left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \;-\; \frac{\mu_{c^q}}{\lambda_{\mathsf{pa}(c^q)}}\right\}$$

Because $W^q_j(\boldsymbol{\mu}, \boldsymbol{\lambda})$ is convex in $\boldsymbol{\mu}$ for each q and $\boldsymbol{\lambda}$ (by Lemma B.2) and $1/\lambda_j - 1/\lambda_{\mathsf{pa}(j)} \geq 0$ for all $j \in \mathsf{T} \setminus \mathcal{N}$, it follows that the function above is convex in $\boldsymbol{\mu}$.

We are left to show that the convexity is strict, for which it is sufficient to show that the following function is strictly convex: $f(\boldsymbol{\mu}) \stackrel{\text{def}}{=} \sum_{q=1}^{Q} W^q_{\mathsf{root}}(\boldsymbol{\mu}, \boldsymbol{\lambda})$. We show strict convexity through a direct application of its definition. The function $f(\cdot)$ is strictly convex if for $\boldsymbol{\mu} \neq \bar{\boldsymbol{\mu}}$ such that $\mu_1 = \bar{\mu}_1 = 0$

and $\theta \in (0,1)$, we have $f(\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}}) < \theta f(\boldsymbol{\mu})+(1-\theta)f(\bar{\boldsymbol{\mu}})$. To arrive at a contradiction, suppose that for some $\boldsymbol{\mu} \neq \bar{\boldsymbol{\mu}}$ and $\theta \in (0,1)$, we have that

$$\begin{aligned} & f(\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}}) = \theta f(\boldsymbol{\mu})+(1-\theta)f(\bar{\boldsymbol{\mu}}) \\ \iff & \sum_{q=1}^{Q} W^q_{\mathsf{root}}(\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}}),\boldsymbol{\lambda}) = \sum_{q=1}^{Q} \theta W^q_{\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}) + (1-\theta)W^q_{\mathsf{root}}(\bar{\boldsymbol{\mu}},\boldsymbol{\lambda}) \ . \end{aligned} \tag{11}$$

By Lemma B.2, for each $q \in \{1,\ldots,Q\}$, $W^q_{\mathsf{root}}(\theta\boldsymbol{\mu}+(1-\theta)\bar{\boldsymbol{\mu}}),\boldsymbol{\lambda}) \leq \theta W^q_{\mathsf{root}}(\boldsymbol{\mu},\boldsymbol{\lambda}) + (1-\theta)W^q_{\mathsf{root}}(\bar{\boldsymbol{\mu}},\boldsymbol{\lambda})$, with equality occurring if and only if $\mu_\ell = \bar{\mu}_\ell + \kappa^q$ for all $\ell \in \mathcal{S}^q$, where $\kappa^q \in \mathbb{R}$ is some constant. Therefore, for Equation (11) to be satisfied, we must have equality occurring for each $q \in \{1,\ldots,Q\}$. As a result, there exists $\kappa^1, \kappa^2, \ldots, \kappa^Q$ such that for each $q \in \{1,\ldots,Q\}$,

$$\mu_\ell = \bar{\mu}_\ell + \kappa^q \text{ for all } \ell \in \mathcal{S}^q.$$

To arrive at a contradiction, we now establish that $\kappa^q = 0$ for all q, which implies that $\boldsymbol{\mu} = \bar{\boldsymbol{\mu}}$ since $\mathcal{N} = \cup_{q\in[Q]}\mathcal{S}^q$. This contradicts our assumption that $\boldsymbol{\mu} \neq \bar{\boldsymbol{\mu}}$!

To show that $\kappa^q = 0$ for all $q = 1,\ldots,Q$, we first prove the following claim.

Claim: $\kappa^1 = \kappa^2 = \cdots = \kappa^Q$.

To prove the claim, it suffices to show that $\kappa^1 = \kappa^2$. Exactly the same argument applies to show that $\kappa^{q_1} = \kappa^{q_2}$ for all $q_1 \neq q_2$. If $\mathcal{S}^1 \cap \mathcal{S}^2 \neq \varnothing$, the result is trivially true because there exists $\ell \in \mathcal{S}^1 \cap \mathcal{S}^2$, which implies that $\kappa^1 = \mu_\ell - \bar{\mu}_\ell = \kappa^2$, which is the desired result. So, suppose that $\mathcal{S}^1 \cap \mathcal{S}^2 = \varnothing$. Pick $\ell_1 \in \mathcal{S}^1$ and $\ell_2 \in \mathcal{S}^2$. Since Comp is strongly connected, there exists a path $\ell_1 = j_0 \to j_1 \to \cdots \to j_{m-1} \to j_m = \ell_2$ in Comp from ℓ_1 to ℓ_2. By our construction of Comp, for each edge (j_{t-1}, j_t), there exists a set $\mathcal{S}^{h_t} \in \{\mathcal{S}^1,\ldots,\mathcal{S}^Q\}$ such that $\{j_{t-1}, j_t\} \subseteq \mathcal{S}^{h_t}$. Therefore, for each $t = 1,\ldots,m$, $j_t \in \mathcal{S}^{h_t} \cap \mathcal{S}^{h_{t+1}}$, and thus, $\kappa^{h_t} = \mu_{j_t} - \bar{\mu}_{j_t} = \kappa^{h_{t+1}}$. Since this is true for all $t = 1,\ldots,m$, it follows that

$$\kappa^{h_1} = \kappa^{h_2} = \cdots = \kappa^{h_m} \ .$$

Since $\ell_1 \in \mathcal{S}^1 \cap \mathcal{S}^{h_1}$ and $\ell_2 \in \mathcal{S}^2 \cap \mathcal{S}^{h_m}$, we have that $\kappa^1 = \kappa^{h_1} = \kappa^{h_m} = \kappa^2$. This is the desired result, proving the claim.

We will now use the above claim to show that $\kappa^q = 0$ for all q. Consider the subset $\mathcal{S}^q$ that contains product 1. Because $\mu_1 = \bar{\mu}_1 = 0$, it follows that $\kappa^q = 0$. Because $\kappa^1 = \ldots \kappa^Q$, it follows that $\kappa^q = 0$ for all $q = 1,\ldots,Q$. This completes the proof of sufficiency.

■

D.2 Proof of Theorem 3.3

Proof: We will show that for each non-leaf node $j \in \mathsf{T} \setminus \mathcal{N}$ such that $j \neq \mathsf{root}$, the function NegLog function varies with respect to λ_j if and only if there exists a transaction $q \in \{1, \ldots, Q\}$ such that node j has at least two children in sub-tree T^q. Consider an arbitrary non-leaf node $j \neq \mathsf{root}$. The sufficiency follows immediately from Lemma B.3. To establish the necessity, assume that NegLog function varies with respect to λ_j. Suppose, on the contrary, that for every transaction q, node j has at most one child in the tree T^q. Then, by definition, for every q,

$$W_j^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \;=\; \lambda_j \log \left(\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda})/\lambda_j} \right) = \sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} W_k(\mathcal{S}; \boldsymbol{\mu}, \boldsymbol{\lambda}) \;,$$

where the last equality follows because there is at most one term in the summand. Therefore, for every q, $W_j^q(\boldsymbol{\mu}, \boldsymbol{\lambda})$ is independent of λ_j, which implies that $-\log \psi^q_{\mathsf{root} \to c^q}(\boldsymbol{\mu}, \boldsymbol{\lambda})$ is independent of λ_j, which implies that NegLog function is also independent of λ_j. Contradiction! Therefore, there must be at least one transaction q such that node j has at least two children in the tree T^q. This proves the necessity. ■

Appendix E: Proofs of Theorem 4.1, Theorem 4.3, Theorem 4.4, Theorem 4.5, Theorem 4.6, and Theorem 4.8

E.1 Proof of Theorem 4.1

The first part of the theorem follows from the Proof of Sufficiency argument in the proof of Theorem 3.2 in Appendix D.1 above.

For the second part, note that from Theorem 2.2, it follows that

$$\begin{aligned}
\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda}) &= \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \frac{W_j^q(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_j} \;-\; \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \frac{W_j^q(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}} \\
&= \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \frac{W_j^q(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_j} \;-\; \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \{\mathsf{root}\}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \frac{W_j^q(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(j)}},
\end{aligned}$$

where the second equality follows since by definition, $\lambda_\ell = +\infty$ for all $\ell \in \mathcal{N}$ and $\lambda_{\mathsf{pa}(\mathsf{root})} = +\infty$. Then, substituting $\lambda_j = e^{-\delta_{[j]}}$ for all $j \in \mathsf{T} \setminus \mathcal{N}$, it follows that $\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}) = F_1(\boldsymbol{\mu}, \boldsymbol{\delta}) - F_2(\boldsymbol{\mu}, \boldsymbol{\delta})$. Next, we need to show that F_1 and F_2 are strictly convex in $\boldsymbol{\delta}$ for any fixed $\boldsymbol{\mu}$. We first show that the function $F_1(\boldsymbol{\mu}, \boldsymbol{\delta})$ is convex in $\boldsymbol{\delta}$ and then establish later that it is strictly convex. By the homogeneity property of the weight function in Lemma B.1 above, we have

$$F_1(\boldsymbol{\mu}, \boldsymbol{\delta}) \;=\; \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} W_j^q(\boldsymbol{\mu}, \boldsymbol{\lambda}(\boldsymbol{\delta})) \cdot e^{\delta_{[j]}} \;=\; \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} W_j^q\left(\boldsymbol{\mu} \cdot e^{\delta_{[j]}}, \boldsymbol{\lambda}(\boldsymbol{\delta}) \cdot e^{\delta_{[j]}}\right) \;,$$

where recall that the vector $\boldsymbol{\lambda}(\boldsymbol{\delta}) = (\exp\left(-\delta_{[j]}\right) : j \in \mathsf{T} \setminus \mathcal{N})$. Then, for each term in the above summation, the mapping $\boldsymbol{\delta} \mapsto W_j^q\left(\boldsymbol{\mu} \cdot e^{\delta_{[j]}}, \boldsymbol{\lambda}(\boldsymbol{\delta}) \cdot e^{\delta_{[j]}}\right)$ is convex because it is a composition of an

increasing convex function $W_j^q(\cdot,\cdot)$ (by Lemmas B.2 and B.3) with a collection of convex functions, $\mu_\ell \cdot e^{\delta_{[j]}} = \mu_\ell \cdot e^{\sum_{k\in \mathsf{path}[\mathsf{root},j]}\delta_k}$, and $e^{-\delta_{[k]}}\cdot e^{\delta_{[j]}} = e^{\sum_{h\in\mathsf{path}[\mathsf{root},j]}\delta_h - \sum_{h\in\mathsf{path}[\mathsf{root},k]}\delta_h}$. Note that $\mu_\ell \cdot e^{\delta_{[j]}}$ and $e^{-\delta_{[k]}}\cdot e^{\delta_{[j]}}$ are convex in $\boldsymbol{\delta}$ because the expressions in the exponents are linear functions of $\boldsymbol{\delta}$. Therefore, $F_1(\boldsymbol{\mu},\boldsymbol{\delta})$ is convex in $\boldsymbol{\delta}$.

To establish strict convexity, we show that one of the terms in the summand is strictly convex. Consider the term corresponding to the root node:

$$f(\boldsymbol{\mu},\boldsymbol{\delta}) \;=\; \sum_{q=1}^{Q} W_{\mathsf{root}}^q\left(\boldsymbol{\mu}\cdot e^{\delta_{[\mathsf{root}]}}, \boldsymbol{\lambda}(\boldsymbol{\delta})\cdot e^{\delta_{[\mathsf{root}]}}\right) \;=\; \sum_{q=1}^{Q} W_{\mathsf{root}}^q(\boldsymbol{\mu},\boldsymbol{\lambda}(\boldsymbol{\delta})) \quad \text{since} \quad \delta_{[\mathsf{root}]} = \delta_{\mathsf{root}} = 0.$$

We now show that $f(\boldsymbol{\mu},\boldsymbol{\delta})$ is strictly convex in $\boldsymbol{\delta}$. For any $\boldsymbol{\delta}\neq\bar{\boldsymbol{\delta}}$ and scalar $x\in(0,1)$, we want to show that

$$f(\boldsymbol{\mu}, x\cdot\boldsymbol{\delta}+(1-x)\cdot\bar{\boldsymbol{\delta}}) \;<\; xf(\boldsymbol{\mu},\boldsymbol{\delta}) + (1-x)f(\boldsymbol{\mu},\bar{\boldsymbol{\delta}}).$$

Since $W_{\mathsf{root}}^{q'}(\boldsymbol{\mu},\boldsymbol{\lambda}(\boldsymbol{\delta}))$ is convex in $\boldsymbol{\delta}$ for all $q'\in\{1,\ldots,Q\}$, it suffices to exhibit one q such that $W_{\mathsf{root}}^q\left(\boldsymbol{\mu},\boldsymbol{\lambda}(x\boldsymbol{\delta}+(1-x)\bar{\boldsymbol{\delta}})\right) \;<\; xW_{\mathsf{root}}^q\left(\boldsymbol{\mu},\boldsymbol{\lambda}(\boldsymbol{\delta})\right) + (1-x)W_{\mathsf{root}}^q\left(\boldsymbol{\mu},\boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right)$.

Now, since $\boldsymbol{\delta}\neq\bar{\boldsymbol{\delta}}$, there must exist a non-leaf node j such that $\delta_{[j]}\neq\bar{\delta}_{[j]}$. From the condition in Theorem 3.3, there exists a $q\in\{1,\ldots,Q\}$ such that j has at least two children in T^q. It then follows from Lemmas B.2 and B.3 that $W_{\mathsf{root}}^q(\boldsymbol{\mu},\boldsymbol{\lambda})$ is convex and increasing in $\boldsymbol{\lambda}$, and strictly increasing in λ_j. Further, because of the strict convexity of the exponential function, we have that $e^{x\cdot(-\delta_{[j]})+(1-x)\cdot(-\bar{\delta}_{[j]})} < xe^{-\delta_{[j]}} + (1-x)e^{-\bar{\delta}_{[j]}}$. Together, these facts imply that

$$\begin{aligned} W_{\mathsf{root}}^q\left(\boldsymbol{\mu},\boldsymbol{\lambda}(x\boldsymbol{\delta}+(1-x)\bar{\boldsymbol{\delta}})\right) \;&<\; W_{\mathsf{root}}^q\left(\boldsymbol{\mu}\;,\; x\boldsymbol{\lambda}(\boldsymbol{\delta})+(1-x)\boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \\ &\leq\; xW_{\mathsf{root}}^q\left(\boldsymbol{\mu},\boldsymbol{\lambda}(\boldsymbol{\delta})\right) + (1-x)W_{\mathsf{root}}^q\left(\boldsymbol{\mu},\boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right)\;. \end{aligned}$$

Consequently, F_1 is strictly convex. The proof of the strict convexity of F_2 follows from an almost identical argument, but with the function f re-defined as $f(\boldsymbol{\mu},\boldsymbol{\delta}) = \sum_{q=1}^{Q}\sum_{j\in\,\mathsf{Children}(\mathsf{root})\cap\mathsf{T}^q} W_j^q(\boldsymbol{\mu},\boldsymbol{\lambda}(\boldsymbol{\delta}))$. ■

E.2 Proof of Theorem 4.3

Before going into the proof, we show that the partial derivatives $(\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\delta})/\partial\delta_k \;:\; k\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\}))$ can be computed efficiently:

Lemma E.1 *For all $k\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})$, it follows that*

$$\frac{\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\delta})}{\partial\delta_k} = -\sum_{j\in\mathsf{T}_k\setminus\mathcal{N}} \lambda_j\cdot\frac{\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial\lambda_j},$$

Proof. The result follows by invoking the multi-variable chain rule of derivatives:

$$\begin{aligned}\frac{\partial \mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\delta})}{\partial \delta_k} &= \sum_{j\in\mathsf{T}\setminus\mathcal{N}} \frac{\partial \lambda_j}{\partial \delta_k}\cdot\frac{\partial \mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \lambda_j}\\ &= \sum_{j\in\mathsf{T}_k\setminus\mathcal{N}} \frac{\partial \lambda_j}{\partial \delta_k}\cdot\frac{\partial \mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \lambda_j}\\ &= \sum_{j\in\mathsf{T}_k\setminus\mathcal{N}} -\lambda_j\cdot\frac{\partial \mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \lambda_j},\end{aligned}$$

where the second inequality follows since $\lambda_j = e^{-\sum_{i\in\mathsf{path}[\mathsf{root},j]}\delta_i}$ and the last since $\frac{\partial\lambda_j}{\partial\delta_k} = -\lambda_j$ for all $j\in\mathsf{T}_k\setminus\mathcal{N}$. The claim then follows. ■

Here is the proof of Theorem 4.3. Define $J(\alpha) := H^{(s)}\left(\left\{\boldsymbol{\delta}^{(s)} - \alpha\cdot\nabla_{\boldsymbol{\delta}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)})\right\}^{+}\right)$ for all $\alpha\in\mathbb{R}_+$. We first state the following lemma that establishes the piecewise convexity of $J(\cdot)$:

Lemma E.2 *For each $j\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})$, denote $d_j = \frac{\partial\mathsf{NegLog}}{\partial\delta_j}\left(\boldsymbol{\mu}^{(s+1)},\boldsymbol{\delta}^{(s)}\right)$ and define t_j as*

$$t_j = \begin{cases} +\infty & \text{if } d_j\le 0\\ \delta_j^{(s)}/d_j & \text{otherwise}\end{cases}.$$

Next, let $0 = t_{(0)} < t_{(1)} < \ldots < t_{(I)} = +\infty$ denote the sorted values in the set $\{t_j\ :\ j\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})\}\cup\{0,+\infty\}$.

Then, for each $i\in\{0,1,\ldots,I-1\}$, the function $J(\alpha)$ is either **constant** *or* **strictly convex** *on the interval $\left[t_{(i)},t_{(i+1)}\right]$. In particular, $J(\cdot)$ is piecewise convex on $\mathbb{R}_+$ with I pieces.*

Proof. Define the vector $\boldsymbol{M}(\alpha) = \left\{\boldsymbol{\delta}^{(s)} - \alpha\boldsymbol{d}\right\}^{+}$ for any $\alpha\ge 0$. Then, for any $\alpha\in\left[t_{(i)},t_{(i+1)}\right]$ and any $j\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})$, it can be verified that

$$M_j(\alpha) = \begin{cases}\delta_j^{(s)} - \alpha\cdot d_j & \text{if } t_{(i+1)}\le t_j\\ 0 & \text{otherwise}\end{cases}. \tag{12}$$

Now, suppose the function $J(\alpha)$ is *not* constant on the interval $\left[t_{(i)},t_{(i+1)}\right]$. In other words, there exists $\bar\alpha,\hat\alpha\in\left[t_{(i)},t_{(i+1)}\right]$ with $\bar\alpha\ne\hat\alpha$ such that $J(\bar\alpha)\ne J(\hat\alpha)$. Since $J(\alpha) = H^{(s)}\left(\boldsymbol{M}(\alpha)\right)$, we must have $\boldsymbol{M}(\bar\alpha)\ne\boldsymbol{M}(\hat\alpha)$. From (12), this further implies that there exists $j_i\in\mathsf{T}\setminus(\mathcal{N}\cup\{\mathsf{root}\})$ such that $d_{j_i}\ne 0$ and $M_{j_i}(\alpha) = \delta_{j_i}^{(s)} - \alpha\cdot d_{j_i}$ for all $\alpha\in\left[t_{(i)},t_{(i+1)}\right]$.

We show that $J(\alpha)$ must be strictly convex on $\left[t_{(i)},t_{(i+1)}\right]$. For that, consider $\alpha_1,\alpha_2\in\left[t_{(i)},t_{(i+1)}\right]$ with $\alpha_1\ne\alpha_2$ and $w\in(0,1)$. From (12), it can be verified that

$$\boldsymbol{M}(w\cdot\alpha_1 + (1-w)\cdot\alpha_2) = w\cdot\boldsymbol{M}(\alpha_1) + (1-w)\cdot\boldsymbol{M}(\alpha_2) \tag{13}$$

Moreover, since $M_{j_i}(\alpha_1)\ne M_{j_i}(\alpha_2)$ it follows that $\boldsymbol{M}(\alpha_1)\ne\boldsymbol{M}(\alpha_2)$. Then, it follows that

$$J\left(w\cdot\alpha_1 + (1-w)\cdot\alpha_2\right) = H^{(s)}\left(\boldsymbol{M}(w\cdot\alpha_1 + (1-w)\cdot\alpha_2)\right)$$

$$
\begin{aligned}
&= H^{(s)}\left(w \cdot \boldsymbol{M}(\alpha_1) + (1-w) \cdot \boldsymbol{M}(\alpha_2)\right) \\
&< w \cdot H^{(s)}\left(\boldsymbol{M}(\alpha_1)\right) + (1-w) \cdot H^{(s)}\left(\boldsymbol{M}(\alpha_2)\right) \\
&= w \cdot J(\alpha_1) + (1-w) \cdot J(\alpha_2),
\end{aligned}
$$

where the second equality follows from (13) and the inequality follows since $\boldsymbol{M}(\alpha_1) \neq \boldsymbol{M}(\alpha_2)$ and the fact that $H^{(s)}(\cdot)$ is strictly convex as established in Lemma 4.9. This establishes the strict convex of $J(\cdot)$ on $\left[t_{(i)}, t_{(i+1)}\right]$.

The result then follows from observing that $\cup_{i=0}^{I-1}\left[t_{(i)}, t_{(i+1)}\right] = \mathbb{R}_+$.

■

To complete the proof of Theorem 4.3, note that the number of pieces I satisfies

$$
\begin{aligned}
I &= \left|\{t_j \ : \ j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})\} \cup \{0, +\infty\}\right| - 1 \\
&\leq \left|\{t_j \ : \ j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})\}\right| + 2 - 1 \\
&= \left|\{t_j \ : \ j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})\}\right| + 1 \\
&\leq |\mathsf{T} \setminus \mathcal{N}|
\end{aligned}
$$

Then, to obtain $\alpha^{(s)}$, we first compute $\alpha_0, \alpha_1, \ldots, \alpha_{I-1}$ as the following:

$$
\alpha_i = \underset{\alpha \in \left[t_{(i)}, t_{(i+1)}\right]}{\arg\min} \ J(\alpha),
$$

which can be done efficiently since $J(\alpha)$ is either constant or strictly convex on each interval $\left[t_{(i)}, t_{(i+1)}\right)$. Finally, it follows that

$$
\alpha^{(s)} = \underset{\alpha \in \left\{\alpha_0, \alpha_1, \ldots, \alpha_{I-1}\right\}}{\arg\min} \ J(\alpha).
$$

E.3 Proof of Theorem 4.4

We begin with the following key lemma that will be useful in the proof.

Lemma E.3 *For any $\boldsymbol{\mu} \in \mathsf{Dom}_1$ and $\boldsymbol{\delta} \in \mathsf{Dom}_2$, it follows that*

$$
\sum_{\ell \in \mathcal{N}} a_\ell\left(\boldsymbol{\mu}, \boldsymbol{\delta}\right) \times \exp\left(\delta_{[\mathsf{pa}(\ell)]}\right) = \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \times \exp\left(\delta_{[\mathsf{pa}(\ell)]}\right),
$$

where $a_\ell(\boldsymbol{\mu}, \boldsymbol{\delta})$ is as defined in (5).

Proof. We leverage the fact that $\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})$ is shift-invariant in the variable $\boldsymbol{\mu}$. Specifically, for any $x \in \mathbb{R}$, we have that

$$
\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}) = \mathsf{NegLog}(\boldsymbol{\mu} + x \cdot \boldsymbol{e}, \boldsymbol{\delta}),
$$

where $\boldsymbol{e} \in \mathbb{R}^N$ denotes the vector of all ones. In other words, for any $\boldsymbol{\mu}$ and $\boldsymbol{\delta}$, the function $g\colon \mathbb{R} \to \mathbb{R}$ defined as $g(x) = \mathsf{NegLog}(\boldsymbol{\mu} + x \cdot \boldsymbol{e}, \boldsymbol{\delta})$ is a constant as a function of x. Therefore, the derivative $g'(x)$ of $g(x)$ with respect to x is zero everywhere.

We can compute the derivative of $g(\cdot)$ using the chain rule as follows:

$$g'(x) = \frac{\mathrm{d}g}{\mathrm{d}x} = \sum_{\ell \in \mathcal{N}} \frac{\partial \mathsf{NegLog}(\boldsymbol{\mu} + x \cdot \boldsymbol{e}, \boldsymbol{\delta})}{\partial \mu_\ell} \frac{\mathrm{d}}{\mathrm{d}x}(\mu_\ell + x) = \sum_{\ell \in \mathcal{N}} \frac{\partial \mathsf{NegLog}(\boldsymbol{\mu} + x \cdot \boldsymbol{e}, \boldsymbol{\delta})}{\partial \mu_\ell}.$$

Equating the above derivative to zero at $x = 0$ yields $\sum_{\ell \in \mathcal{N}} \partial \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}) / \partial \mu_\ell = 0$. We now use the definition of the constant $a_\ell(\boldsymbol{\mu}, \boldsymbol{\delta})$ for each $\ell \in \mathcal{N}$ to obtain

$$0 = \sum_{\ell \in \mathcal{N}} \frac{\partial \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta})}{\partial \mu_\ell} = \sum_{\ell \in \mathcal{N}} \left(a_\ell(\boldsymbol{\mu}, \boldsymbol{\delta}) - \mathsf{sales}_\ell\right) \cdot \exp\left(\delta_{[\mathsf{pa}(\ell)]}\right)$$

The result of the lemma now follows. ■

We are now ready to prove Theorem 4.4. Below, we denote $\boldsymbol{\lambda}^{(s)} = \boldsymbol{\lambda}(\boldsymbol{\delta}^{(s)})$ to simplify certain expressions. In Theorem 4.8, we show that $\sum_{\ell \in \mathcal{N}} G_\ell\left(\mu_\ell \,|\, \boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}\right)$ is a majorizing surrogate [see Definition 4.7] for the mapping $\boldsymbol{\mu} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}^{(s)})$ at $\boldsymbol{\mu}^{(s)}$. From this, it follows that for all $\boldsymbol{\mu} \in \mathsf{Dom}_1$:

$$\begin{aligned}
\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}^{(s)}) &\le \sum_{\ell \in \mathcal{N}} G_\ell\left(\mu_\ell \,|\, \boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}\right) \\
&= \sum_{\ell \in \mathcal{N}} \mathsf{C}_\ell(\boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}) + \sum_{\ell \in \mathcal{N}} a_\ell\left(\boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}\right) \exp\left(\frac{\mu_\ell - \mu_\ell^{(s)}}{\lambda^{(s)}_{\mathsf{pa}(\ell)}}\right) - \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \cdot \left(\frac{\mu_\ell - \mu_\ell^{(s)}}{\lambda^{(s)}_{\mathsf{pa}(\ell)}}\right)
\end{aligned} \tag{14}$$

and,

$$\mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) = \sum_{\ell \in \mathcal{N}} \mathsf{C}_\ell(\boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}) + \sum_{\ell \in \mathcal{N}} a_\ell\left(\boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}\right) \tag{15}$$

Combining equations (14) and (15), we obtain that for all $\boldsymbol{\mu} \in \mathsf{Dom}_1$:

$$\begin{aligned}
&\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
&\le \sum_{\ell \in \mathcal{N}} a_\ell\left(\boldsymbol{\mu}^{(s)},\ \boldsymbol{\delta}^{(s)}\right) \left(\exp\left(\frac{\mu_\ell - \mu_\ell^{(s)}}{\lambda^{(s)}_{\mathsf{pa}(\ell)}}\right) - 1\right) - \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \cdot \left(\frac{\mu_\ell - \mu_\ell^{(s)}}{\lambda^{(s)}_{\mathsf{pa}(\ell)}}\right)
\end{aligned}$$

Now, noting that $\tilde{\mu}_\ell^{(s+1)} = \mu_\ell^{(s)} + \lambda^{(s)}_{\mathsf{pa}(\ell)} \cdot \log\left(\mathsf{sales}_\ell / a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})\right)$ for all $\ell \in \mathcal{N}$, we get (we show in the proof of Theorem 4.8 below that $a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) > 0$ for all s)

$$\begin{aligned}
&\mathsf{NegLog}(\tilde{\boldsymbol{\mu}}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
&\le \sum_{\ell \in \mathcal{N}} a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})} - 1\right) - \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right) \\
&= \sum_{\ell \in \mathcal{N}} \left(\mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \mathsf{sales}_\ell \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right)\right).
\end{aligned} \tag{16}$$

We first establish that in each iteration, the MM update makes a non-positive improvement (recall that we are minimizing the objective function) by showing that the upper bound in equation (16) above is always non-positive. In fact, we establish a stronger result that each term in the upper bound summation is non-positive; that is, $\mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \mathsf{sales}_\ell \cdot \log\left(\mathsf{sales}_\ell / a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})\right) \leq 0$ for all $\ell \in \mathcal{N}$.

To that end, for each $\ell \in \mathcal{N}$, note that $\tilde{\mu}_\ell^{(s+1)}$ is the minimizer of $G_\ell(\cdot \,|\, \boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})$ by definition. Therefore, we have that

$$
\begin{aligned}
& G_\ell(\tilde{\mu}_\ell^{(s+1)} \,|\, \boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \leq G_\ell(\mu_\ell^{(s)} \,|\, \boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
\Longrightarrow\; & a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \exp\left(\frac{\tilde{\mu}_\ell^{(s+1)} - \mu_\ell^{(s)}}{\lambda_{\mathsf{pa}(\ell)}^{(s)}}\right) - \mathsf{sales}_\ell \cdot \frac{\tilde{\mu}_\ell^{(s+1)} - \mu_\ell^{(s)}}{\lambda_{\mathsf{pa}(\ell)}^{(s)}} \leq a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
\Longrightarrow\; & a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})} - \mathsf{sales}_\ell \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right) \leq a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
\Longrightarrow\; & \mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \mathsf{sales}_\ell \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right) \leq 0.
\end{aligned}
$$

It now follows from the arguments above that

$$
\begin{aligned}
& \mathsf{NegLog}(\tilde{\boldsymbol{\mu}}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
& \leq \sum_{\ell \in \mathcal{N}} \left(\mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \mathsf{sales}_\ell \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right)\right) \\
& \leq \sum_{\ell \in \mathcal{N}} \frac{\lambda_{\mathsf{lower}}}{\lambda_{\mathsf{pa}(\ell)}^{(s)}} \left[\mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \mathsf{sales}_\ell \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right)\right] \\
& = \lambda_{\mathsf{lower}} \cdot \left(\sum_{\ell \in \mathcal{N}} \frac{\mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}{\lambda_{\mathsf{pa}(\ell)}^{(s)}} \right) - \lambda_{\mathsf{lower}} \cdot \sum_{\ell \in \mathcal{N}} \frac{\mathsf{sales}_\ell}{\lambda_{\mathsf{pa}(\ell)}^{(s)}} \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right) \\
& = \lambda_{\mathsf{lower}} \cdot \left(\sum_{\ell \in \mathcal{N}} \left\{ \mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\} \times \exp\left(\delta_{[\mathsf{pa}(\ell)]}^{(s)}\right) \right) - \lambda_{\mathsf{lower}} \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \cdot \exp\left(\delta_{[\mathsf{pa}(\ell)]}^{(s)}\right) \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right) \\
& = -\lambda_{\mathsf{lower}} \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \cdot \exp\left(\delta_{[\mathsf{pa}(\ell)]}^{(s)}\right) \cdot \log\left(\frac{\mathsf{sales}_\ell}{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right),
\end{aligned}
$$

where the first inequality follows because $\lambda_{\mathsf{pa}(\ell)}^{(s)} = e^{-\sum_{k \in \mathsf{path}[\mathsf{root}, \mathsf{pa}(\ell)]} \delta_k^{(s)}} \geq e^{-(\,\mathsf{height}(\mathsf{root}) - 1) \times \delta_{\mathsf{upper}}} := \lambda_{\mathsf{lower}}$ for all $\ell \in \mathcal{N}$ and the last equality follows from the result of Lemma E.3 which implies $\sum_{\ell \in \mathcal{N}} \left\{ \mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\} \times \exp\left(\delta_{[\mathsf{pa}(\ell)]}^{(s)}\right) = 0$.

The last expression on the right hand side above has the form of KL-divergence between two probability distributions, except that the corresponding terms in the expressions do not sum to 1. To address that, we normalize $(a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \colon \ell \in \mathcal{N})$ and $(\mathsf{sales}_\ell \colon \ell \in \mathcal{N})$ as follows. We let

$T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})$ denote the sums $\sum_{\ell \in \mathcal{N}} a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \times \exp\left(\delta^{(s)}_{[\mathsf{pa}(\ell)]}\right) = \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \times \exp\left(\delta^{(s)}_{[\mathsf{pa}(\ell)]}\right)$ and define

$$\overline{a}_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) = \frac{a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \cdot \exp\left(\delta^{(s)}_{[\mathsf{pa}(\ell)]}\right)}{T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}; \quad \overline{\mathsf{sales}}_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) = \frac{\mathsf{sales}_\ell \cdot \exp\left(\delta^{(s)}_{[\mathsf{pa}(\ell)]}\right)}{T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}.$$

It is clear from our definitions that $\overline{\boldsymbol{a}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) = \left(\overline{a}_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \colon \ell \in \mathcal{N}\right)$ and $\overline{\mathsf{sales}} = \left(\overline{\mathsf{sales}}_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \colon \ell \in \mathcal{N}\right)$ are valid distributions over the set $\mathcal{N}$. It now follows from these definitions that

$$\begin{aligned}
&\mathsf{NegLog}(\tilde{\boldsymbol{\mu}}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
&\leq -T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \cdot \lambda_{\mathsf{lower}} \sum_{\ell \in \mathcal{N}} \overline{\mathsf{sales}}_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \log\left(\frac{\overline{\mathsf{sales}}_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}{\overline{a}_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}\right) \\
&= -T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \cdot \lambda_{\mathsf{lower}} \cdot D_{KL}\left(\overline{\mathsf{sales}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \| \overline{\boldsymbol{a}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})\right).
\end{aligned}$$

We can simplify the above expression further by invoking Pinsker's inequality, which states that $D_{KL}\left(\overline{\mathsf{sales}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \| \overline{\boldsymbol{a}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})\right) \geq 1/2 \left\| \overline{\mathsf{sales}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \overline{\boldsymbol{a}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_1^2$. We now obtain

$$\begin{aligned}
&\mathsf{NegLog}(\tilde{\boldsymbol{\mu}}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \\
&\quad \leq -T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \cdot \lambda_{\mathsf{lower}} \cdot \frac{1}{2} \left\| \overline{\mathsf{sales}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \overline{\boldsymbol{a}}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_1^2 \\
&\quad = -\frac{\lambda_{\mathsf{lower}}}{2 \cdot T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})} \left(\sum_{\ell \in \mathcal{N}} \left| \left\{ \mathsf{sales}_\ell - a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\} \times \exp\left(\delta^{(s)}_{[\mathsf{pa}(\ell)]}\right) \right| \right)^2 \\
&\quad = -\frac{\lambda_{\mathsf{lower}}}{2 \cdot T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})} \left(\sum_{\ell \in \mathcal{N}} \left| \frac{\partial \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})}{\partial \mu_\ell} \right| \right)^2 = -\frac{\lambda_{\mathsf{lower}}}{2 \cdot T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})} \left\| \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_1^2,
\end{aligned}$$

where the second equality follows from the definition of $a_\ell(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)})$.

To complete the proof, we note that

$$T(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) = \sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell \times \cdot \exp\left(\delta^{(s)}_{[\mathsf{pa}(\ell)]}\right) = \sum_{\ell \in \mathcal{N}} \frac{\mathsf{sales}_\ell}{\lambda^{(s)}_{\mathsf{pa}(\ell)}} \leq \sum_{\ell \in \mathcal{N}} \frac{\mathsf{sales}_\ell}{\lambda_{\mathsf{lower}}} = \frac{\sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell}{\lambda_{\mathsf{lower}}} = \frac{1}{\lambda_{\mathsf{lower}}}.$$

The inequality follows because $\lambda^{(s)}_{\mathsf{pa}(\ell)} \geq \lambda_{\mathsf{lower}}$ for all $\ell \in \mathcal{N}$. The last equality follows from the definition of sales_ℓ, which denotes the fraction of sales of product ℓ in the dataset, so that $\sum_{\ell \in \mathcal{N}} \mathsf{sales}_\ell = 1$.

Putting everything together, we obtain

$$\mathsf{NegLog}(\tilde{\boldsymbol{\mu}}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \leq \frac{-\lambda^2_{\mathsf{lower}}}{2} \left\| \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_1^2.$$

The improvement bound for the MM update follows from observing that $\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)}) = \mathsf{NegLog}(\tilde{\boldsymbol{\mu}}^{(s+1)}, \boldsymbol{\delta}^{(s)})$ since NegLog is shift-invariant w.r.t. $\boldsymbol{\mu}$. $\square$

E.4 Proof of Theorem 4.5

We begin with the following lemma:

Lemma E.4 *If $\lambda_j \geq \lambda_{\mathsf{lower}}$ for all $j \in \mathsf{T} \setminus \mathcal{N}$, then the mapping $\boldsymbol{\mu} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda})$ is L-smooth with $L \leq 1/\lambda_{\mathsf{lower}}^2$. Moreover, the upper bound $1/\lambda_{\mathsf{lower}}^2$ is tight, upto constant factors.*

Proof. We use the definition of smoothness stated in (Bubeck 2015, Section 3.2). A continuously differentiable function $f\colon \mathbb{R}^n \to \mathbb{R}$ is L-smooth if its gradient ∇f is L-Lipschitz continuous, i.e.

$$\|\nabla f(\boldsymbol{x}) - \nabla f(\boldsymbol{y})\|_2 \leq L \|\boldsymbol{x} - \boldsymbol{y}\|_2 \quad \forall \boldsymbol{x}, \boldsymbol{y} \in \mathbb{R}^n$$

For a twice continuously differentiable function, the above condition is equivalent to $\nabla^2 f(\boldsymbol{x}) \preceq L\boldsymbol{I}_n$ for all $\boldsymbol{x} \in \mathbb{R}^n$, where $\boldsymbol{I}_n$ is the $n \times n$ identity matrix. In other words, all the eigenvalues of the Hessian are bounded above by L. We will use this definition to derive the smoothness constant L.

In our context, $f(\boldsymbol{\mu}) = \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda})$. We use the fact that the trace (sum of diagonal elements) of a matrix is equal to the sum of its eigenvalues. Consider the sum of the diagonal entries of the Hessian matrix $\nabla^2 f(\boldsymbol{\mu})$:

$$\begin{aligned}
&\sum_{\ell \in \mathcal{N}} \frac{\partial^2 f(\boldsymbol{\mu})}{\partial \mu_\ell^2} \\
&= \sum_{\ell \in \mathcal{N}} \frac{\partial}{\partial \mu_\ell} \left(\frac{\partial f(\boldsymbol{\mu})}{\partial \mu_\ell} \right) \\
&\overset{\text{(a)}}{=} \sum_{\ell \in \mathcal{N}} \frac{\partial}{\partial \mu_\ell} \left(\frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}} \right) \cdot \psi_{j \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) - \frac{\mathsf{sales}_\ell}{\lambda_{\mathsf{pa}(\ell)}} \right) \\
&= \sum_{\ell \in \mathcal{N}} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}} \right) \cdot \frac{\partial}{\partial \mu_\ell} \psi_{j \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \\
&\overset{\text{(b)}}{=} \sum_{\ell \in \mathcal{N}} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}} \right) \cdot \frac{\partial}{\partial \mu_\ell} \prod_{k \in \mathsf{path}(j,\ell]} \psi_{\mathsf{pa}(k) \to k}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \\
&\overset{\text{(c)}}{=} \sum_{\ell \in \mathcal{N}} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}} \right) \cdot \psi_{j \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \cdot \left(\sum_{k \in \mathsf{path}(j,\ell]} \frac{\partial \psi_{\mathsf{pa}(k) \to k}^q(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\partial \mu_\ell} \Big/ \psi_{\mathsf{pa}(k) \to k}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \right) \\
&\overset{\text{(d)}}{=} \sum_{\ell \in \mathcal{N}} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}} \right) \cdot \psi_{j \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \cdot \left(\sum_{k \in \mathsf{path}(j,\ell]} \frac{\psi_{k \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) - \psi_{\mathsf{pa}(k) \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_{\mathsf{pa}(k)}} \right) \\
&\overset{\text{(e)}}{\leq} \sum_{\ell \in \mathcal{N}} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}} \right) \cdot \psi_{j \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \cdot \left(\sum_{k \in \mathsf{path}(j,\ell]} \frac{\psi_{k \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) - \psi_{\mathsf{pa}(k) \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda})}{\lambda_{\mathsf{lower}}} \right) \\
&= \sum_{\ell \in \mathcal{N}} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}} \right) \cdot \psi_{j \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \cdot \left(\psi_{\ell \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) - \psi_{j \to \ell}^q(\boldsymbol{\mu}, \boldsymbol{\lambda}) \right) \Big/ \lambda_{\mathsf{lower}}
\end{aligned}$$

$$\overset{\text{(f)}}{\leq} \sum_{\ell\in\mathcal{N}} \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j\in\mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \cdot \psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) \,/\, \lambda_{\mathsf{lower}}$$

$$= \frac{1}{Q} \sum_{q=1}^{Q} \sum_{j\in\mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right) \cdot \frac{\sum_{\ell\in\mathcal{N}} \psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\lambda_{\mathsf{lower}}}$$

$$\overset{\text{(g)}}{=} \frac{1}{Q\cdot\lambda_{\mathsf{lower}}} \sum_{q=1}^{Q} \sum_{j\in\mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q\in\mathsf{T}_j\}} \cdot \left(\frac{1}{\lambda_j} - \frac{1}{\lambda_{\mathsf{pa}(j)}}\right)$$

$$\overset{\text{(h)}}{=} \frac{1}{Q\cdot\lambda_{\mathsf{lower}}} \sum_{q=1}^{Q} \frac{1}{\lambda_{\mathsf{pa}(c^q)}}$$

$$\overset{\text{(i)}}{\leq} \frac{1}{\lambda^2_{\mathsf{lower}}}$$

where the justifications for the equalities and inequalities in (a) - (i) are given below.

(a) The equality follows from the expression for $\frac{\partial \mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \mu_\ell}$ from Lemma C.2 with $\boldsymbol{\lambda} = \boldsymbol{\lambda}(\boldsymbol{\delta})$.

(b) The equality since $\psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) = \prod_{k\in\mathsf{path}(j,\ell]} \psi^q_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda})$.

(c) The equality follows from the product rule of derivatives.

(d) The equality follows since $\psi^q_{\mathsf{pa}(k)\to k}(\boldsymbol{\mu},\boldsymbol{\lambda}) = e^{-\left(W^q_{\mathsf{pa}(k)}(\boldsymbol{\mu},\boldsymbol{\lambda}) - W^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})\right)/\lambda_{\mathsf{pa}(k)}}$ and $\frac{\partial W^q_k(\boldsymbol{\mu},\boldsymbol{\lambda})}{\partial \mu_\ell} = \psi^q_{k\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})$ for all $k \in \mathsf{T}\setminus\mathcal{N}$ from Lemma B.4.

(e) The inequality follows since $\lambda_j \geq \lambda_{\mathsf{lower}}$ for all $j \in \mathsf{T}\setminus\mathcal{N}$ by assumption and $\psi^q_{k\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) \geq \psi^q_{\mathsf{pa}(k)\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda})$ for all $k \in \mathsf{path}(j,\ell]$.

(f) The equality follows since $\psi^q_{\ell\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) - \psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) \leq 1$.

(g) The equality follows since $\sum_{\ell\in\mathcal{N}} \psi^q_{j\to\ell}(\boldsymbol{\mu},\boldsymbol{\lambda}) = 1$ for any j such that $c^q \in \mathsf{T}_j$.

(h) The equality follows since $\frac{1}{\lambda_{\mathsf{pa}(\mathsf{root})}} = +\infty$.

(i) The inequality follows since $\lambda_{\mathsf{pa}(c^q)} \geq \lambda_{\mathsf{lower}}$ by assumption.

Since $f(\boldsymbol{\mu})$ is convex on $\mathbb{R}^n$ (see proof of Theorem 4.1 above), all the eigenvalues of the Hessian $\nabla^2 f(\boldsymbol{\mu})$ are non-negative, for any $\boldsymbol{\mu} \in \mathbb{R}^n$. Then, it follows that:

$$\begin{aligned}
\text{max eigenvalue of } \nabla^2 f(\boldsymbol{\mu}) &\leq \text{sum of eignvalues of } \nabla^2 f(\boldsymbol{\mu}) \\
&= \text{trace of } \nabla^2 f(\boldsymbol{\mu}) \\
&= \sum_{\ell\in\mathcal{N}} \frac{\partial^2 f(\boldsymbol{\mu})}{\partial \mu^2_\ell} \\
&\leq \frac{1}{\lambda^2_{\mathsf{lower}}}
\end{aligned}$$

This establishes the smoothness of $f(\boldsymbol{\mu})$.

To show that the upper bound $1/\lambda^2_{\mathsf{lower}}$ is tight up to constant factors, we consider the special case of the MNL model, i.e. $\mathsf{T} = \mathcal{N} \cup \{\mathsf{root}\}$. Further, we consider a special case of the MNL model, the Bradley-Terry model, in which each offer-set contains exactly two products. For the Bradley-Terry

model, Vojnovic et al. (2020, Lemma 3.1) showed that the negative log-likelihood is L-smooth on $\mathbb{R}^n$, where $L = \lambda_{\max}(\mathbf{L_M})/4Q$.[12] Here, $\mathbf{M}$ is an $n \times n$ matrix with $M_{i,j}$ being the number of times offer-set $\{i,j\}$ is observed in the transaction data, and $\lambda_{\max}(\mathbf{L_M})$ is the largest eigenvalue of the Laplacian matrix $\mathbf{L_M} = \mathbf{D_M} - \mathbf{M}$, where $\mathbf{D_M}$ is a diagonal matrix whose diagonal elements are the row sums of $\mathbf{M}$.

Now, consider the following transaction dataset consisting of $Q = 2 \cdot (n-1)$ transactions:

$$(\mathcal{S}^q, c^q) = \begin{cases} (\{1, q+1\}, q+1) & \text{for } 1 \le q \le n-1 \\ (\{1, q-n+2\}, 1) & \text{for } n \le q \le Q \end{cases}$$

For the above dataset, it is easy to check that the product co-occurrence matrix $\mathbf{M} = 2\mathbf{M}'$ where $\mathbf{M}'$ is the adjacency matrix of the comparison graph Comp defined in Definition 3.1. Since each edge in Comp appears in both directions, we can treat Comp as an undirected graph. Now, since the degree of the node corresponding to product 1 in Comp is equal to $n-1$, it follows from Zhang (2011, Theorem 3.19) that $\lambda_{\max}(\mathbf{L_{M'}}) = n$. Then, it follows $L = \lambda_{\max}(\mathbf{L_M})/4Q = 2\lambda_{\max}(\mathbf{L_{M'}})/4Q = 2n/4Q = \frac{n}{4\cdot(n-1)}$. For large n, we have $L \approx 1/4$. Finally, note that $\lambda_{\mathsf{lower}} = 1$ for the MNL model. This shows that the upper bound $1/\lambda_{\mathsf{lower}}^2$ is tight up to constant factors. □

We are now ready to prove Theorem 4.5. Since $\delta_j^{(s)} \le \delta_{\mathsf{upper}}$ for all $j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})$ by assumption, it follows that $\lambda_j^{(s)} \ge \lambda_{\mathsf{lower}}$ for all $j \in \mathsf{T} \setminus \mathcal{N}$, where $\boldsymbol{\lambda}^{(s)} = \boldsymbol{\lambda}(\boldsymbol{\delta}^{(s)})$. Then, it follows from Lemma E.4 that $\mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\delta}^{(s)})$ is L-smooth. We then leverage the standard quadratic upper bound property (Bubeck 2015, Lemma 3.4) for smooth functions to obtain:

$$\begin{aligned}
&\mathsf{NegLog}(\boldsymbol{\mu}_{\mathsf{GD}}^{(s+1)}, \boldsymbol{\delta}^{(s)}) \\
&\le \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) + \left\langle \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}), \boldsymbol{\mu}_{\mathsf{GD}}^{(s+1)} - \boldsymbol{\mu}^{(s)} \right\rangle + \frac{L}{2} \left\| \boldsymbol{\mu}_{\mathsf{GD}}^{(s+1)} - \boldsymbol{\mu}^{(s)} \right\|_2^2 \\
&= \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \frac{1}{L} \left\langle \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}), \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\rangle + \frac{1}{2L} \left\| \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_2^2 \\
&\text{(using the definition of } \boldsymbol{\mu}_{\mathsf{GD}}^{(s+1)}) \\
&= \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) - \frac{1}{2L} \left\| \nabla_{\boldsymbol{\mu}} \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) \right\|_2^2
\end{aligned}$$

■

E.5 Proof of Theorem 4.6

We first establish that the constant D is finite, for which it is sufficient to show that the set $\mathcal{U} = \left\{ \boldsymbol{\mu} \in \mathsf{Dom}_1 : \mathsf{NegLog}(\boldsymbol{\mu}) \le \mathsf{NegLog}(\boldsymbol{\mu}^{(0)}) \right\}$ is bounded. From the Proof of Sufficiency argument

[12] Vojnovic et al. (2020) consider the un-normalized negative log-likelihood and therefore do not have the normalization by the total number of transactions Q in their result.

in the proof of Theorem 3.2 in Appendix D.1 above, it follows that for all $\boldsymbol{\mu} \in \mathsf{Dom}_1$ (since $\mathsf{Dom}_1 = \mathcal{D}_1$):

$$\mathsf{NegLog}(\boldsymbol{\mu}) \geq \|\boldsymbol{\mu}\|_\infty b^*/Q,$$

where b^* is as defined in Appendix D.1 and $b^* > 0$ by Lemma D.1. Therefore, it follows that for any $\boldsymbol{\mu} \in \mathcal{U}$:

$$\mathsf{NegLog}(\boldsymbol{\mu}) \leq \mathsf{NegLog}(\boldsymbol{\mu}^{(0)}) \implies \|\boldsymbol{\mu}\|_\infty \leq \frac{\mathsf{NegLog}(\boldsymbol{\mu}^{(0)}) \cdot Q}{b^*},$$

and therefore the set $\mathcal{U}$ is bounded.

Next, we derive the convergence rates. We begin with the guarantee for the MM algorithm. The proof follows from existing results, see e.g., Allen-Zhu and Orecchia (2014, Fact B.1 in Appendix B). We reproduce the arguments here for completeness. For each $s' \geq 0$, define $\mathsf{NegLogGap}^{(s')} := \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) - \mathsf{NegLog}(\boldsymbol{\mu}^*)$. Note that $\mathsf{NegLogGap}^{(s')} \geq 0$ for all $s' \geq 0$. Moreover, since the result is trivially true for any $s \geq 1$ such that $\mathsf{NegLogGap}^{(s)} = 0$, we suppose that $\mathsf{NegLogGap}^{(s')} > 0$ for all $0 \leq s' \leq s$.

Now, recall that $\mathsf{NegLog}(\boldsymbol{\mu})$ is convex in $\boldsymbol{\mu}$. Then, it follows that for any $s' \geq 0$:

$$\begin{aligned}
\mathsf{NegLogGap}^{(s')} &= \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) - \mathsf{NegLog}(\boldsymbol{\mu}^*) \\
&\leq \left\langle \nabla \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}), \boldsymbol{\mu}^{(s')} - \boldsymbol{\mu}^* \right\rangle \\
&\leq \left\| \nabla \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) \right\|_1 \cdot \left\| \boldsymbol{\mu}^{(s')} - \boldsymbol{\mu}^* \right\|_\infty \\
&\leq \left\| \nabla \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) \right\|_1 \cdot D
\end{aligned} \tag{17}$$

where the first inequality follows from the subgradient inequality for convex functions (note that we write the gradient as $\nabla \mathsf{NegLog}$ instead of $\nabla_{\boldsymbol{\mu}} \mathsf{NegLog}$), the second follows from the Holder's inequality, and the final follows since the MM algorithm guarantees an improving solution in each iteration and using the definition of D in the statement of the theorem.

Next, by plugging in $\lambda_{\mathsf{lower}} = 1$ in the improvement bound from Theorem 4.4, it follows that for all $0 \leq s' < s$:

$$\begin{aligned}
&\mathsf{NegLog}(\boldsymbol{\mu}^{(s'+1)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) \leq -\frac{1}{2} \left\| \nabla \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) \right\|_1^2 \\
\implies &\mathsf{NegLogGap}^{(s'+1)} - \mathsf{NegLogGap}^{(s')} \leq -\frac{1}{2} \left\| \nabla \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) \right\|_1^2 \\
\implies &\mathsf{NegLogGap}^{(s')} - \mathsf{NegLogGap}^{(s'+1)} \geq \frac{1}{2} \left\| \nabla \mathsf{NegLog}(\boldsymbol{\mu}^{(s')}) \right\|_1^2 \\
\implies &\mathsf{NegLogGap}^{(s')} - \mathsf{NegLogGap}^{(s'+1)} \geq \frac{\left(\mathsf{NegLogGap}^{(s')}\right)^2}{2D^2} \\
\implies &\frac{\mathsf{NegLogGap}^{(s')} - \mathsf{NegLogGap}^{(s'+1)}}{\mathsf{NegLogGap}^{(s')} \cdot \mathsf{NegLogGap}^{(s'+1)}} \geq \frac{\mathsf{NegLogGap}^{(s')}}{2D^2 \cdot \mathsf{NegLogGap}^{(s'+1)}}
\end{aligned}$$

$$\implies \frac{1}{\mathsf{NegLogGap}^{(s'+1)}} - \frac{1}{\mathsf{NegLogGap}^{(s')}} \geq \frac{\mathsf{NegLogGap}^{(s')}}{2D^2 \cdot \mathsf{NegLogGap}^{(s'+1)}}$$
$$\implies \frac{1}{\mathsf{NegLogGap}^{(s'+1)}} - \frac{1}{\mathsf{NegLogGap}^{(s')}} \geq \frac{1}{2D^2},$$

where the third implication follows from (17), the fourth follows from diving both sides by $\mathsf{NegLogGap}^{(s')} \cdot \mathsf{NegLogGap}^{(s'+1)}$, and the final follows since $\mathsf{NegLogGap}^{(s')} \geq \mathsf{NegLogGap}^{(s'+1)}$ as the MM algorithm guarantees an improving solution in each iteration.

Summing the last inequality above from $s'=0$ to $s-1$ (note that we have $s \geq 1$), it follows that

$$\sum_{s'=0}^{s-1} \left(\frac{1}{\mathsf{NegLogGap}^{(s'+1)}} - \frac{1}{\mathsf{NegLogGap}^{(s')}} \right) \geq \sum_{s'=0}^{s-1} \frac{1}{2D^2}$$
$$\implies \frac{1}{\mathsf{NegLogGap}^{(s)}} - \frac{1}{\mathsf{NegLogGap}^{(0)}} \geq \frac{s}{2D^2}$$
$$\implies \frac{1}{\mathsf{NegLogGap}^{(s)}} \geq \frac{s}{2D^2}$$
$$\implies \mathsf{NegLogGap}^{(s)} \leq \frac{2D^2}{s}$$
$$\implies \mathsf{NegLog}(\boldsymbol{\mu}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^*) \leq \frac{2D^2}{s}$$

The result then follows.

For the GD algorithm, starting from the improvement bound in Theorem 4.5 and using the fact that GD with step size $1/L$ also guarantees an improving solution in each iteration, the above sequence of arguments can be repeated to show that for all $s \geq 1$

$$\mathsf{NegLog}(\boldsymbol{\mu}_{\mathsf{GD}}^{(s)}) - \mathsf{NegLog}(\boldsymbol{\mu}^*) \leq \frac{2LnD^2}{s}$$

The result then follows.

E.6 Proof of Theorem 4.8

The proof of the theorem makes use of the following lemma:

Lemma E.5 *Given any $\bar{\boldsymbol{\delta}} \in \mathsf{Dom}_2$, for each $q = 1, \ldots, Q$ and each nonleaf node $j \in \mathsf{T}^q \setminus \mathcal{S}^q$, a majorizing surrogate for the function $\boldsymbol{\mu} \mapsto \frac{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q(\boldsymbol{\mu},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})) \times \exp(\bar{\delta}_{[j]})}}{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q(\bar{\boldsymbol{\mu}},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})) \times \exp(\bar{\delta}_{[j]})}}$ at $\bar{\boldsymbol{\mu}}$ is given by the following separable function:*

$$\boldsymbol{\mu} \;\mapsto\; \mathsf{C}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) \;+\; \exp\left(\bar{\delta}_{[j]}\right) \sum_{\ell \in \mathcal{N}} \exp\left(-\bar{\delta}_{[\mathsf{pa}(\ell)]}\right) \times \psi_{j \to \ell}^q\left(\bar{\boldsymbol{\mu}},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times e^{(\mu_\ell - \bar{\mu}_\ell) \times \exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)} \;,$$

where $\mathsf{C}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}})$ is a constant depending only on $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}})$ that is irrelevant for our optimization.

Proof. Fix an arbitrary q and let $\bar{\boldsymbol{\lambda}} = \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})$. We will prove the result by induction on the height of j. For the base case, suppose j has a height of one, that is, it is a parent of some leaf node. In this case, note that for each $\ell \in \mathsf{Children}(j)$, $\exp\left(\bar{\delta}_{[j]}\right) \times \exp\left(-\bar{\delta}_{[\mathsf{pa}(\ell)]}\right) = 1$. Then,

$$
\begin{aligned}
&\frac{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q\left(\boldsymbol{\mu},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times \exp\left(\bar{\delta}_{[j]}\right)}}{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q\left(\bar{\boldsymbol{\mu}},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times \exp\left(-\bar{\delta}_{[j]}\right)}} \\
&\overset{\text{(a)}}{=} \frac{\sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}^q} e^{\mu_\ell / \bar{\lambda}_j}}{\sum_{i \in \mathsf{Children}(j) \cap \mathcal{S}^q} e^{\bar{\mu}_i / \bar{\lambda}_j}} \\
&= \sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}^q} \frac{e^{\bar{\mu}_\ell / \bar{\lambda}_j}}{\sum_{i \in \mathsf{Children}(j) \cap \mathcal{S}^q} e^{\bar{\mu}_i / \bar{\lambda}_j}} \times e^{(\mu_\ell - \bar{\mu}_\ell)/\bar{\lambda}_{\mathsf{pa}(\ell)}} \\
&= \sum_{\ell \in \mathsf{Children}(j) \cap \mathcal{S}^q} \psi_{j \to \ell}^q\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) \times e^{(\mu_\ell - \bar{\mu}_\ell)/\bar{\lambda}_{\mathsf{pa}(\ell)}} \\
&\overset{\text{(b)}}{=} \sum_{\ell \in \mathcal{N}} \psi_{j \to \ell}^q\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) \times e^{(\mu_\ell - \bar{\mu}_\ell)/\bar{\lambda}_{\mathsf{pa}(\ell)}} \\
&= \sum_{\ell \in \mathcal{N}} \psi_{j \to \ell}^q\left(\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times e^{(\mu_\ell - \bar{\mu}_\ell) \times \exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)}\ ,
\end{aligned}
$$

where (a) follows from the definition of the weight function at leaf nodes and because $\bar{\boldsymbol{\lambda}} = \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})$, and (b) follows because if $\ell \notin \mathsf{Children}(j) \cap \mathcal{S}^q$, then $\ell \notin \mathsf{T}^q$, so $\psi_{j \to \ell}^q\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) = 0$. This completes the base case.

To establish the induction step, suppose that the result holds for all nodes k of height at most H. In other words, for every nonleaf node $k \in \mathsf{T}^q$ of height at most H, there is a constant C_k that depend only on $\bar{\boldsymbol{\mu}}$ and $\bar{\boldsymbol{\delta}}$ such that for all $\boldsymbol{\mu}$,

$$
\begin{aligned}
&\frac{\sum_{m \in \mathsf{Children}(k) \cap \mathsf{T}^q} e^{W_m^q\left(\boldsymbol{\mu},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times \exp\left(\bar{\delta}_{[k]}\right)}}{\sum_{m \in \mathsf{Children}(k) \cap \mathsf{T}^q} e^{W_m^q\left(\bar{\boldsymbol{\mu}},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times \exp\left(\bar{\delta}_{[k]}\right)}} \\
&\quad \le\ \mathsf{C}_k\ +\ \exp\left(\bar{\delta}_{[k]}\right) \sum_{\ell \in \mathcal{N}} \exp\left(-\bar{\delta}_{[\mathsf{pa}(\ell)]}\right) \times \psi_{k \to \ell}^q\left(\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times e^{(\mu_\ell - \bar{\mu}_\ell) \times \exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)}\ ,
\end{aligned}
$$

and the above inequality is tight at $\boldsymbol{\mu} = \bar{\boldsymbol{\mu}}$. Now, consider a nonleaf node j of height $H+1$. Letting C denote a constant that depend only on $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}})$, we have

$$
\begin{aligned}
&\frac{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q\left(\boldsymbol{\mu},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times \exp\left(\bar{\delta}_{[j]}\right)}}{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q\left(\bar{\boldsymbol{\mu}},\, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \times \exp\left(\bar{\delta}_{[j]}\right)}} \\
&\overset{\text{(a)}}{=} \frac{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q\left(\boldsymbol{\mu},\, \bar{\boldsymbol{\lambda}}\right) / \bar{\lambda}_j}}{\sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_k^q\left(\bar{\boldsymbol{\mu}},\, \bar{\boldsymbol{\lambda}}\right) / \bar{\lambda}_j}} \\
&\overset{\text{(b)}}{=} \sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} \frac{\left(\sum_{m \in \mathsf{Children}(k) \cap \mathsf{T}^q} e^{W_m^q(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}})/\bar{\lambda}_k}\right)^{\bar{\lambda}_k/\bar{\lambda}_j}}{\sum_{s \in \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W_s^q\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right)/\bar{\lambda}_j}}
\end{aligned}
$$

$$
\begin{aligned}
&\overset{\text{(c)}}{\leq} \mathsf{C} + \sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}^q} \frac{\bar\lambda_k}{\bar\lambda_j} \times \frac{\left(\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda})/\bar\lambda_k}\right)^{\lambda_k/\lambda_j}}{\sum_{s\in \mathsf{Children}(j)\cap \mathsf{T}^q} e^{W_s^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right)/\bar\lambda_j}} \times \frac{\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\boldsymbol\mu,\bar{\boldsymbol\lambda})/\bar\lambda_k}}{\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda})/\bar\lambda_k}} \\
&\overset{\text{(d)}}{=} \mathsf{C} + \sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}^q} \frac{\bar\lambda_k}{\bar\lambda_j} \times \frac{e^{W_k^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda})/\bar\lambda_j}}{\sum_{s\in \mathsf{Children}(j)\cap \mathsf{T}^q} e^{W_s^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right)/\bar\lambda_j}} \times \frac{\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\boldsymbol\mu,\bar{\boldsymbol\lambda})/\bar\lambda_k}}{\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda})/\bar\lambda_k}} \\
&\overset{\text{(e)}}{=} \mathsf{C} + \sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}^q} \frac{\bar\lambda_k}{\bar\lambda_j} \times \psi_{j\to k}^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}) \times \frac{\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\boldsymbol\mu,\bar{\boldsymbol\lambda})/\bar\lambda_k}}{\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda})/\bar\lambda_k}} \\
&\overset{\text{(f)}}{\leq} \mathsf{C} + \sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}^q} \mathsf{C}_k + \left(\frac{\bar\lambda_k}{\bar\lambda_j} \times \psi_{j\to k}^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}) \times \frac{1}{\bar\lambda_k} \sum_{\ell\in\mathcal{N}} \bar\lambda_{\mathsf{pa}(\ell)} \times \psi_{k\to\ell}^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right) \times e^{(\mu_\ell-\bar\mu_\ell)/\bar\lambda_{\mathsf{pa}(\ell)}}\right) \\
&\overset{\text{(g)}}{=} \mathsf{C}_j + \sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}^q} \frac{1}{\bar\lambda_j} \times \psi_{j\to k}^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}) \sum_{\ell\in\mathcal{S}^q\cap \mathsf{T}_k^q} \bar\lambda_{\mathsf{pa}(\ell)} \times \psi_{k\to\ell}^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right) \times e^{(\mu_\ell-\bar\mu_\ell)/\bar\lambda_{\mathsf{pa}(\ell)}} \\
&\overset{\text{(h)}}{=} \mathsf{C}_j + \frac{1}{\bar\lambda_j} \sum_{\ell\in\mathcal{S}^q\cap \mathsf{T}_j^q} \bar\lambda_{\mathsf{pa}(\ell)} \times \psi_{j\to\ell}^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right) \times e^{(\mu_\ell-\bar\mu_\ell)/\bar\lambda_{\mathsf{pa}(\ell)}} \\
&\overset{\text{(i)}}{=} \mathsf{C}_j + \frac{1}{\bar\lambda_j} \sum_{\ell\in\mathcal{N}} \bar\lambda_{\mathsf{pa}(\ell)} \times \psi_{j\to\ell}^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right) \times e^{(\mu_\ell-\bar\mu_\ell)/\bar\lambda_{\mathsf{pa}(\ell)}} \\
&\overset{\text{(j)}}{=} \mathsf{C}_j + \exp\left(\bar\delta_{[j]}\right) \sum_{\ell\in\mathcal{N}} \exp\left(-\bar\delta_{[\mathsf{pa}(\ell)]}\right) \times \psi_{j\to\ell}^q\left(\bar{\boldsymbol\mu},\boldsymbol\lambda(\bar{\boldsymbol\delta})\right) \times e^{(\mu_\ell-\bar\mu_\ell)\times\exp\left(\bar\delta_{[\mathsf{pa}(\ell)]}\right)},
\end{aligned}
$$

where the justifications for the equalities and inequalities in (a) - (j) are given below.

(a) The equality follows because $\bar{\boldsymbol\lambda} = \boldsymbol\lambda(\bar{\boldsymbol\delta})$.

(b) The equality follows from the definition that for each $k \in \mathsf{Children}(j) \cap \mathsf{T}^q$, $W_k^q\left(\boldsymbol\mu,\bar{\boldsymbol\lambda}\right) = \bar\lambda_k \log\left(\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\boldsymbol\mu,\bar{\boldsymbol\lambda})/\bar\lambda_k}\right)$, so

$$
e^{W_k^q\left(\boldsymbol\mu,\bar{\boldsymbol\lambda}\right)/\bar\lambda_j} = \left(\sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\boldsymbol\mu,\bar{\boldsymbol\lambda})/\bar\lambda_k}\right)^{\bar\lambda_k/\bar\lambda_j}.
$$

(c) For the inequality (c), we apply the following sub-gradient inequality: for all $\alpha \in (0,1]$ and $x, y \in \mathbb{R}_+$, $x^\alpha \leq y^\alpha + \alpha y^{\alpha-1}(x-y) = (1-\alpha)y^\alpha + \alpha y^\alpha \frac{x}{y}$, with equality if and only if $x = y$. The inequality (c) follows from the application of the sub-gradient inequality where for each $k \in \mathsf{Children}(k) \cap \mathsf{T}^q$, we let

$$
x_k = \sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\boldsymbol\mu,\bar{\boldsymbol\lambda})/\bar\lambda_k}, \qquad y_k = \sum_{m\in \mathsf{Children}(k)\cap \mathsf{T}^q} e^{W_m^q(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda})/\bar\lambda_k}, \qquad \text{and} \qquad \alpha_k = \frac{\bar\lambda_k}{\bar\lambda_j},
$$

and note that $\alpha_k \in (0,1]$ because $\bar\lambda_k \leq \bar\lambda_j$ since $k \in \mathsf{Children}(j)$. Also, the constant C corresponds to $\sum_{k\in \mathsf{Children}(j)\cap \mathsf{T}^q}(1-\alpha_k)y_k^{\alpha_k}$, which only depends on $\bar{\boldsymbol\mu}$ and $\bar{\boldsymbol\lambda}$.

(d) This follows from the definition of $e^{W_k^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right)/\bar\lambda_j}$.

(e) This equality follows from the definition of $\psi_{j\to k}^q\left(\bar{\boldsymbol\mu},\bar{\boldsymbol\lambda}\right)$.

(f) The inequality (f) follows from an application of the inductive hypothesis to the children of j, and these children have height of at most H, and from collecting terms.

(g) This follows from defining $\mathsf{C}_j = \mathsf{C} + \sum_{k \in \mathsf{Children}(j) \cap \mathsf{T}^q} \mathsf{C}_k$ and use the fact that $\psi^q_{k\to\ell}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}) = 0$ for all $\ell \notin \mathcal{S}^q \cap \mathsf{T}^q_k$.

(h) The equality (h) follows because $\psi^q_{j\to\ell}\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) = \psi^q_{j\to k}\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) \times \psi^q_{k\to\ell}\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right)$.

(i) The equality (i) follows by noting that for all $\ell \notin \mathcal{S}^q \cap \mathsf{T}^q_j$, $\psi^q_{j\to\ell}(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}) = 0$.

(j) Finally, the last equality (j) again follows because $\bar{\boldsymbol{\lambda}} = \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})$.

This completes the induction step. Therefore, the result holds for all nodes j. ■

We are now ready to prove the theorem. For each $\ell \in \mathcal{N}$, define $G_\ell(\cdot \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}}) \colon \mathbb{R} \to \mathbb{R}$ as

$$G_\ell(x \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}}) \ = \ e^{(x-\bar{\mu}_\ell)\times \exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)} \cdot a_\ell(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) \ - \ (x - \bar{\mu}_\ell) \cdot \exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right) \cdot \mathsf{sales}_\ell \tag{18}$$

Then, letting $\bar{\boldsymbol{\lambda}} = \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})$, it follows from Theorem 2.2 that

$$\mathsf{NegLog}(\boldsymbol{\mu}, \bar{\boldsymbol{\delta}}) \ = \ \frac{1}{Q}\sum_{q=1}^{Q} \left\{ W^q_{\mathsf{root}}\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right) \ + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q)} W^q_j\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right) \cdot \left(1/\bar{\lambda}_j - 1/\bar{\lambda}_{\mathsf{pa}(j)}\right) \ - \ \mu_{c^q} \,/\, \bar{\lambda}_{\mathsf{pa}(c^q)} \right\} .$$

For each nonleaf node $j \in \mathsf{T} \setminus \mathcal{N}$, let $\bar{\zeta}_j = (1/\bar{\lambda}_j) - (1/\bar{\lambda}_{\mathsf{pa}(j)})$ and we set $\bar{\zeta}_{\mathsf{root}} = 1$. Note that $\bar{\zeta}_j \geq 0$ for all j because $0 < \bar{\lambda}_j \leq \bar{\lambda}_{\mathsf{pa}(j)}$. Then, we have

$$\begin{aligned}
&\mathsf{NegLog}(\boldsymbol{\mu}, \bar{\boldsymbol{\delta}}) \\
&= \ \frac{1}{Q}\sum_{q=1}^{Q} \left\{ W^q_{\mathsf{root}}\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right) \ + \sum_{j \in \mathsf{path}(\mathsf{root}, c^q)} \bar{\zeta}_j\, W^q_j(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}) \right\} \ - \ \frac{1}{Q}\sum_{q=1}^{Q} \mu_{c^q} \,/\, \bar{\lambda}_{\mathsf{pa}(c^q)} \\
&\overset{\text{(a)}}{=} \ \frac{1}{Q}\sum_{q=1}^{Q} \sum_{j \in \mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\left\{c^q \,\in\, \mathsf{T}_j\right\}} \bar{\zeta}_j\, W^q_j\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right) \ - \ \frac{1}{Q}\sum_{\ell \in \mathcal{N}} \left(\mu_\ell \,/\, \bar{\lambda}_{\mathsf{pa}(\ell)}\right) \cdot \left(\sum_{q=1}^{Q} \mathbb{1}_{\{c^q = \ell\}}\right) \\
&\overset{\text{(b)}}{=} \ \frac{1}{Q}\sum_{q=1}^{Q} \sum_{j \in \mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\left\{c^q \,\in\, \mathsf{T}_j\right\}} \bar{\zeta}_j\, W^q_j\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right) \ - \ \sum_{\ell \in \mathcal{N}} (\mu_\ell - \bar{\mu}_\ell) \cdot \mathsf{sales}_\ell \,/\, \bar{\lambda}_{\mathsf{pa}(\ell)} \ - \ \sum_{\ell \in \mathcal{N}} \left(\bar{\mu}_\ell \,/\, \bar{\lambda}_{\mathsf{pa}(\ell)}\right) \cdot \mathsf{sales}_\ell \ ,
\end{aligned}$$

where the equality (a) follows because, for each $j \in \mathsf{T} \setminus \mathcal{N}$, $c^q \in \mathsf{T}_j$ if and only if $j \in \mathsf{path}[\mathsf{root}, c^q)$ and equality (b) follows from the definition of sales_ℓ.

Let $H(\boldsymbol{\mu}) = \frac{1}{Q}\sum_{q=1}^{Q} \sum_{j \in \mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\left\{c^q \,\in\, \mathsf{T}_j\right\}} \bar{\zeta}_j\, W^q_j\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right)$. We will apply Lemma E.5 to construct a majorizing surrogate function for $H(\boldsymbol{\mu})$. Let C denote a constant that depend only on $(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}})$. Then,

$$\begin{aligned}
&H\left(\boldsymbol{\mu}\right) \\
&\overset{\text{(c)}}{=} \ \frac{1}{Q}\sum_{q=1}^{Q} \sum_{j \in \mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}}\, \bar{\zeta}_j\, \bar{\lambda}_j \log\left(\sum_{k \in\, \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W^q_k\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right) / \bar{\lambda}_j}\right) \\
&\overset{\text{(d)}}{\leq} \ \mathsf{C} + \frac{1}{Q}\sum_{q=1}^{Q} \sum_{j \in \mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}}\, \bar{\zeta}_j\, \bar{\lambda}_j \frac{\sum_{k \in\, \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W^q_k\left(\boldsymbol{\mu}, \bar{\boldsymbol{\lambda}}\right) / \bar{\lambda}_j}}{\sum_{k \in\, \mathsf{Children}(j) \cap \mathsf{T}^q} e^{W^q_k\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) / \bar{\lambda}_j}} \\
&\overset{\text{(e)}}{\leq} \ \mathsf{C} + \frac{1}{Q}\sum_{q=1}^{Q} \sum_{j \in \mathsf{T}\setminus\mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}}\, \bar{\zeta}_j\, \bar{\lambda}_j \times \frac{1}{\bar{\lambda}_j} \sum_{\ell \in \mathcal{N}} \bar{\lambda}_{\mathsf{pa}(\ell)} \times \psi^q_{j\to\ell}\left(\bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\lambda}}\right) \times e^{(\mu_\ell - \bar{\mu}_\ell) \,/\, \bar{\lambda}_{\mathsf{pa}(\ell)}}
\end{aligned}$$

$$\overset{\text{(f)}}{=}\ \mathsf{C}+\frac{1}{Q}\sum_{\ell\in\mathcal{N}} e^{(\mu_\ell-\bar{\mu}_\ell)}\big/\,\bar{\lambda}_{\mathsf{pa}(\ell)}\times\left[\bar{\lambda}_{\mathsf{pa}(\ell)}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\mathcal{N}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\,\bar{\zeta}_j\,\psi^q_{j\to\ell}\left(\bar{\boldsymbol{\mu}},\,\bar{\boldsymbol{\lambda}}\right)\right]$$

$$\overset{\text{(g)}}{=}\ \mathsf{C}+\frac{1}{Q}\sum_{\ell\in\mathcal{N}} e^{(\mu_\ell-\bar{\mu}_\ell)}\big/\,\bar{\lambda}_{\mathsf{pa}(\ell)}\times\left[\bar{\lambda}_{\mathsf{pa}(\ell)}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\mathcal{N}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(1/\bar{\lambda}_j-1/\bar{\lambda}_{\mathsf{pa}(j)}\right)\,\psi^q_{j\to\ell}\left(\bar{\boldsymbol{\mu}},\,\bar{\boldsymbol{\lambda}}\right)\right]$$

where the justifications for the equalities and inequalities in (c) - (g) are given below.

(c) The equality follows from the definition of $W_j^q\left(\boldsymbol{\mu},\,\bar{\boldsymbol{\lambda}}\right)$.

(d) The inequality follows from applying the subgradient inequality for logarithm, which shows that for all $x\in\mathbb{R}_+$ and $y\in\mathbb{R}_+$, $\log(x)\le\log(y)+\frac{1}{y}(x-y)=\frac{x}{y}+\log(y)-1$, with equality if and only if $x=y$. Here, for each $q=1,\ldots,Q$ and $j\in\mathsf{T}\setminus\mathcal{N}$,

$$x_j^q=\sum_{k\in\,\mathsf{Children}(j)} e^{W_k^q\left(\boldsymbol{\mu},\,\bar{\boldsymbol{\lambda}}\right)/\,\bar{\lambda}_j}\qquad\text{and}\qquad y_j^q=\sum_{k\in\,\mathsf{Children}(j)} e^{W_k^q\left(\bar{\boldsymbol{\mu}},\,\bar{\boldsymbol{\lambda}}\right)/\,\bar{\lambda}_j}$$

and the constant $\mathsf{C}=\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}}\left(\log(y_j^q)-1\right)$, which only depends on $(\bar{\boldsymbol{\mu}},\bar{\boldsymbol{\lambda}})$.

(e) The inequality follows from Lemma E.5, and the constant C is updated accordingly.

(f) The equality follows from algebra and rearranging terms.

(g) The equality follows from the definition of $\bar{\zeta}_j$ and the fact that $\bar{\lambda}_{\mathsf{pa(root)}}=+\infty$.

Putting everything together, we have the following majorizing surrogate for $\boldsymbol{\mu}\mapsto\mathsf{NegLog}(\boldsymbol{\mu},\bar{\boldsymbol{\delta}})$:

$$\begin{aligned}
&\mathsf{C}\ +\ \frac{1}{Q}\sum_{\ell\in\mathcal{N}} e^{(\mu_\ell-\bar{\mu}_\ell)}\big/\,\bar{\lambda}_{\mathsf{pa}(\ell)}\times\left[\bar{\lambda}_{\mathsf{pa}(\ell)}\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\mathcal{N}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(1/\bar{\lambda}_j-1/\bar{\lambda}_{\mathsf{pa}(j)}\right)\,\psi^q_{j\to\ell}\left(\bar{\boldsymbol{\mu}},\,\bar{\boldsymbol{\lambda}}\right)\right]\\
&\qquad\qquad-\ \sum_{\ell\in\mathcal{N}}(\mu_\ell\ -\ \bar{\mu}_\ell)\cdot\mathsf{sales}_\ell\big/\,\bar{\lambda}_{\mathsf{pa}(\ell)}\\
&=\ \mathsf{C}+\frac{1}{Q}\sum_{\ell\in\mathcal{N}} e^{(\mu_\ell-\bar{\mu}_\ell)\times\exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)}\times\left[\exp\left(-\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)\sum_{q=1}^{Q}\sum_{j\in\mathsf{T}\setminus\mathcal{N}}\mathbb{1}_{\{c^q\in\mathsf{T}_j\}}\left(\exp\left(\bar{\delta}_{[j]}\right)-\exp\left(\bar{\delta}_{[\mathsf{pa}(j)]}\right)\right)\,\psi^q_{j\to\ell}\left(\bar{\boldsymbol{\mu}},\,\boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right)\right]\\
&\qquad\qquad-\ \sum_{\ell\in\mathcal{N}}(\mu_\ell\ -\ \bar{\mu}_\ell)\times\exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)\times\mathsf{sales}_\ell\\
&=\ \mathsf{C}+\sum_{\ell\in\mathcal{N}} e^{(\mu_\ell-\bar{\mu}_\ell)\times\exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)}\times\exp\left(-\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)\times\left[\left.\frac{\partial\mathsf{NegLog}(\boldsymbol{\mu},\boldsymbol{\delta})}{\partial\mu_\ell}\right|_{\boldsymbol{\mu}=\bar{\boldsymbol{\mu}},\,\boldsymbol{\delta}=\bar{\boldsymbol{\delta}}}+\mathsf{sales}_\ell\cdot\exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)\right]\\
&\qquad\qquad-\ \sum_{\ell\in\mathcal{N}}(\mu_\ell\ -\ \bar{\mu}_\ell)\times\exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)\times\mathsf{sales}_\ell\\
&=\ \mathsf{C}+\sum_{\ell\in\mathcal{N}} e^{(\mu_\ell-\bar{\mu}_\ell)\times\exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)}\cdot a_\ell(\bar{\boldsymbol{\mu}},\bar{\boldsymbol{\delta}})-\sum_{\ell\in\mathcal{N}}(\mu_\ell\ -\ \bar{\mu}_\ell)\times\exp\left(\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)\times\mathsf{sales}_\ell\\
&=\ \mathsf{C}+\sum_{\ell\in\mathcal{N}} G_\ell(\mu_\ell\,|\,\bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}})
\end{aligned}$$

where the first equality follows since $\bar{\boldsymbol{\lambda}}=\boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})$, the second follows from Lemma C.2, the third from the definition of $a_\ell\left(\bar{\boldsymbol{\mu}},\bar{\boldsymbol{\delta}}\right)$, and the final from the definition of $G_\ell(\mu_\ell\,|\,\bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}})$ in (18).

To show that $G_\ell(\cdot \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}})$ is strictly convex, it suffices to show that $a_\ell(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}) > 0$. Using the expression for $\frac{\partial \mathsf{NegLog}(\boldsymbol{\mu}, \boldsymbol{\lambda}(\boldsymbol{\delta}))}{\partial \mu_\ell}$ from Lemma C.2, it follows that

$$\begin{aligned}
a_\ell\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\delta}}\right) &= \frac{\exp\left(-\bar{\delta}_{[\mathsf{pa}(\ell)]}\right)}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \left(\exp\left(\bar{\delta}_{[j]}\right) - \exp\left(\bar{\delta}_{[\mathsf{pa}(j)]}\right)\right) \psi^q_{j \to \ell}\left(\bar{\boldsymbol{\mu}}, \boldsymbol{\lambda}(\bar{\boldsymbol{\delta}})\right) \\
&= \frac{\bar{\lambda}_{\mathsf{pa}(\ell)}}{Q} \sum_{q=1}^{Q} \sum_{j \in \mathsf{T} \setminus \mathcal{N}} \mathbb{1}_{\{c^q \in \mathsf{T}_j\}} \left(1/\bar{\lambda}_j - 1/\bar{\lambda}_{\mathsf{pa}(j)}\right) \psi^q_{j \to \ell}\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) \\
&\geq \frac{\bar{\lambda}_{\mathsf{pa}(\ell)}}{Q} \sum_{q=1}^{Q} \mathbb{1}_{\{c^q \in \mathsf{T}_{\mathsf{root}}\}} \left(1/\bar{\lambda}_{\mathsf{root}} - 1/\bar{\lambda}_{\mathsf{pa}(\mathsf{root})}\right) \psi^q_{\mathsf{root} \to \ell}\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) \\
&= \frac{\bar{\lambda}_{\mathsf{pa}(\ell)}}{Q} \sum_{q=1}^{Q} \psi^q_{\mathsf{root} \to \ell}\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) \\
&> 0,
\end{aligned}$$

where the first inequality follows since each term inside the summation is non-negative, the last equality follows because $\mathbb{1}_{\{c^q \in \mathsf{T}_{\mathsf{root}}\}} = 1$, $\bar{\lambda}_{\mathsf{root}} = 1$ and $\bar{\lambda}_{\mathsf{pa}(\mathsf{root})} = +\infty$, and the final inequality follows because $\bar{\lambda}_{\mathsf{pa}(\ell)} > 0$ and the identifiability condition in Theorem 3.2 ensures that $\ell \in \mathcal{S}^{\hat{q}}$ for some $\hat{q}$ so that $\psi^{\hat{q}}_{\mathsf{root} \to \ell}\left(\bar{\boldsymbol{\mu}}, \bar{\boldsymbol{\lambda}}\right) > 0$.

Finally, the expression for the minimizer follows from setting $\partial G_\ell(\mu_\ell \,|\, \bar{\boldsymbol{\mu}},\ \bar{\boldsymbol{\delta}})/\partial \mu_\ell = 0$.

Appendix F: Improvements via MM and PGD updates for $\boldsymbol{\delta}$

As mentioned in the main text, the improvement guarantee for Step 2 of the A-MM algorithm follows from existing results. In particular, it follows from the standard guarantee for the projected gradient descent (PGD) algorithm for constrained smooth problems, which we reproduce here for completeness. Consider the optimization problem

$$\min_{\boldsymbol{x} \in \mathcal{X}} f(\boldsymbol{x}), \tag{19}$$

where f is L-smooth on $\mathcal{X}$ and $\mathcal{X} \subset \mathbb{R}^d$ is a convex set. We define the *gradient mapping* (Nesterov 2013, Section 2.2.3) of f over $\mathcal{X}$, denoted by $\mathcal{G}_{f,\mathcal{X}} \colon \mathcal{X} \to \mathbb{R}^d$ as

$$\mathcal{G}_{f,\mathcal{X}}(\boldsymbol{x}) = L \cdot \left(\boldsymbol{x} - \mathsf{Proj}_{\mathcal{X}}\left(\boldsymbol{x} - \frac{1}{L}\nabla f(\boldsymbol{x})\right)\right), \tag{20}$$

where $\mathsf{Proj}_{\mathcal{X}}(\cdot)$ is the projection operator onto $\mathcal{X}$, i.e. $\mathsf{Proj}_{\mathcal{X}}(\bar{\boldsymbol{x}}) = \arg\min_{\boldsymbol{x} \in \mathcal{X}} \|\boldsymbol{x} - \bar{\boldsymbol{x}}\|_2^2$ for all $\bar{\boldsymbol{x}} \in \mathbb{R}^d$. The gradient mapping plays the role of the gradient in unconstrained optimization problems since it can be shown that $\boldsymbol{x}^*$ is a stationary point for problem (19) if and only if $\|\mathcal{G}_{f,\mathcal{X}}(\boldsymbol{x}^*)\|_2 = 0$.

Now, starting from an initial solution $\boldsymbol{x}^{(0)} \in \mathcal{X}$, consider solving problem (19) using PGD with step size $1/L$, that is, for each $s \geq 0$:

$$\boldsymbol{x}^{(s+1)} = \mathsf{Proj}_{\mathcal{X}}\left(\boldsymbol{x}^{(s)} - \frac{1}{L}\nabla f(\boldsymbol{x}^{(s)})\right)$$

Then, using the quadratic upper bound for smooth functions, it follows that

$$f(\boldsymbol{x}^{(s+1)}) \leq f(\boldsymbol{x}^{(s)}) + \left\langle \nabla f(\boldsymbol{x}^{(s)}), \boldsymbol{x}^{(s+1)} - \boldsymbol{x}^{(s)} \right\rangle + \frac{L}{2} \left\| \boldsymbol{x}^{(s+1)} - \boldsymbol{x}^{(s)} \right\|_2^2 \tag{21}$$

Next, for any $\boldsymbol{x} \in \mathcal{X}$, denote $D(\boldsymbol{x}) = \left\| \boldsymbol{x} - \left(\boldsymbol{x}^{(s)} - \frac{1}{L} \nabla f(\boldsymbol{x}^{(s)}) \right) \right\|_2^2$ and note that $\boldsymbol{x}^{(s+1)} = \arg\min_{\boldsymbol{x} \in \mathcal{X}} D(\boldsymbol{x})$. The optimality of $\boldsymbol{x}^{(s+1)}$ implies that there is no descent direction from $\boldsymbol{x}^{(s+1)}$:

$$\left\langle \nabla D(\boldsymbol{x}^{(s+1)}), \boldsymbol{x} - \boldsymbol{x}^{(s+1)} \right\rangle \geq 0 \quad \forall\ \boldsymbol{x} \in \mathcal{X}$$

By plugging in the gradient of $D(\cdot)$ and choosing $\boldsymbol{x} = \boldsymbol{x}^{(s)}$, it follows that

$$\left\langle \nabla f(\boldsymbol{x}^{(s)}), \boldsymbol{x}^{(s+1)} - \boldsymbol{x}^{(s)} \right\rangle \leq -L \left\| \boldsymbol{x}^{(s+1)} - \boldsymbol{x}^{(s)} \right\|_2^2$$

Substituting the above in (21), it follows that

$$\begin{aligned} f(\boldsymbol{x}^{(s+1)}) &\leq f(\boldsymbol{x}^{(s)}) - L \left\| \boldsymbol{x}^{(s+1)} - \boldsymbol{x}^{(s)} \right\|_2^2 + \frac{L}{2} \left\| \boldsymbol{x}^{(s+1)} - \boldsymbol{x}^{(s)} \right\|_2^2 \\ &= f(\boldsymbol{x}^{(s)}) - \frac{L}{2} \left\| \boldsymbol{x}^{(s+1)} - \boldsymbol{x}^{(s)} \right\|_2^2 \\ &= f(\boldsymbol{x}^{(s)}) - \frac{1}{2L} \left\| \mathcal{G}_{f,\mathcal{X}}(\boldsymbol{x}^{(s)}) \right\|_2^2, \end{aligned} \tag{22}$$

where the final equality follows from the definition of $\mathcal{G}_{f,\mathcal{X}}(\boldsymbol{x}^{(s)})$.

We adapt the above improvement guarantee to our context. Define the domain $\overline{\mathsf{Dom}_2}$ as:

$$\overline{\mathsf{Dom}_2} = \left\{ \boldsymbol{\delta} \in \mathbb{R}^{|\mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\})|} \ :\ 0 \leq \delta_j \leq \delta_{\mathsf{upper}}\ \forall\ j \in \mathsf{T} \setminus (\mathcal{N} \cup \{\mathsf{root}\}) \right\} \tag{23}$$

Then, we can establish the following:

Lemma F.1 (Improvements via MM and PGD updates for $\boldsymbol{\delta}$) *Suppose $\boldsymbol{\delta}^{(s)} \in \overline{\mathsf{Dom}_2}$ and let $\gamma, \gamma_{\mathsf{GD}} > 0$ denote the smoothness constants of the mappings $\boldsymbol{\delta} \mapsto H\left(\boldsymbol{\delta} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)}\right)$ and $\boldsymbol{\delta} \mapsto \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta})$, respectively, over the domain $\overline{\mathsf{Dom}_2}$. If $\boldsymbol{\delta}^{(s+1)} \in \overline{\mathsf{Dom}_2}$, then the MM update guarantees the following improvement in the negative log-likelihood objective:*

$$\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s+1)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)}) \leq -\frac{1}{2\gamma} \left\| \mathcal{G}_{H, \overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s)}) \right\|_2^2$$

Similarly, let $\boldsymbol{\delta}_{\mathsf{GD}}^{(s+1)}$ denote the PGD update with step size $1/\gamma_{\mathsf{GD}}$. If $\boldsymbol{\delta}_{\mathsf{GD}}^{(s+1)} \in \overline{\mathsf{Dom}_2}$, then the PGD update guarantees the following improvement in the negative log-likelihood objective:

$$\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}_{\mathsf{GD}}^{(s+1)}) - \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)}) \leq -\frac{1}{2\gamma_{\mathsf{GD}}} \left\| \mathcal{G}_{\mathsf{NegLog}, \overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s)}) \right\|_2^2$$

Proof. We start with the MM update. First, it can be shown—using an argument analogous to that in the proof of Theorem 4.4 above—that the eigenvalues of the Hessian matrix $\nabla^2 H(\boldsymbol{\delta} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)})$ are bounded above by a constant $\gamma > 0$ that depends on δ_{upper}, for all $\boldsymbol{\delta} \in \overline{\mathsf{Dom}_2}$. This ensures that $H(\boldsymbol{\delta} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)})$ is γ-smooth on $\overline{\mathsf{Dom}_2}$. Now, define $\bar{\boldsymbol{\delta}}^{(s+1)} \in \overline{\mathsf{Dom}_2}$ as follows:

$$\bar{\boldsymbol{\delta}}^{(s+1)} = \mathsf{Proj}_{\overline{\mathsf{Dom}_2}}\left(\boldsymbol{\delta}^{(s)} - \frac{1}{\gamma}\nabla H(\boldsymbol{\delta}^{(s)} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)})\right),$$

Then, it follows from (22) that

$$\begin{aligned} H(\bar{\boldsymbol{\delta}}^{(s+1)} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)}) &\leq H(\boldsymbol{\delta}^{(s)} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)}) - \frac{1}{2\gamma}\left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s)})\right\|_2^2 \\ &= \mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)}) - \frac{1}{2\gamma}\left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s)})\right\|_2^2, \end{aligned}$$

where the equality follows since $H(\boldsymbol{\delta} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)})$ is a majorizing surrogate. The improvement bound then follows since

$$\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s+1)}) \leq H(\boldsymbol{\delta}^{(s+1)} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)}) \leq H(\bar{\boldsymbol{\delta}}^{(s+1)} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)}),$$

where the second inequality follows since $\nabla H(\boldsymbol{\delta}^{(s)} \,|\, \boldsymbol{\mu}^{(s+1)},\ \boldsymbol{\delta}^{(s)}) = \nabla_{\boldsymbol{\delta}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s)})$ and the fact that the MM update $\boldsymbol{\delta}^{(s+1)} \in \overline{\mathsf{Dom}_2}$ corresponds to the optimal step size.

The improvement bound for the GD update $\boldsymbol{\delta}_{\mathsf{GD}}^{(s+1)}$ follows in an identical fashion by leveraging the smoothness of $\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta})$. In particular, we can show that $\mathsf{NegLog}(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta})$ is γ_{GD}-smooth on $\overline{\mathsf{Dom}_2}$, for some constant $\gamma_{\mathsf{GD}} > 0$ that depends on δ_{upper}. The result then follows from (22). ■

Appendix G: Sublinear convergence rate of the A-MM algorithm

As stated in the main text, the improvements bounds from Theorem 4.4 and Lemma F.1 can be combined to establish a sublinear rate of convergence of the A-MM algorithm, as shown in the following theorem, where we leverage the notations introduced above:

Theorem G.1 (Sublinear convergence of A-MM to a stationary point) *Suppose that* $\boldsymbol{\delta}^{(s)} \in \overline{\mathsf{Dom}_2}$ *for all* $s \geq 0$. *Then, the sequence of iterates* $\left((\boldsymbol{\mu}^{(s)}, \boldsymbol{\delta}^{(s)}) :\ s \geq 0\right)$ *converges to a stationary point of the* MLE problem. *Moreover, we can establish the following guarantee for the iterates:*

$$\min_{0 \leq s' \leq s}\left[\left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')}\right)\right\|_1^2 + \left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s')})\right\|_2^2\right] \ \leq\ \frac{2 \cdot \left(\mathsf{NegLog}(\boldsymbol{\mu}^{(0)}, \boldsymbol{\delta}^{(0)}) - \mathsf{NegLog}^*\right)}{R \cdot (s+1)},$$

where NegLog^* *is the optimal objective for the* MLE problem, *and* $R = \min\left(\lambda_{\mathsf{lower}}^2, 1/\gamma\right)$.

Proof. The first part of the theorem follows from the fact that the A-MM algorithm guarantees an improving solution as long as the current solution is not a stationary point.

For the second part, we leverage the improvements from Theorem 4.4 and Lemma F.1 to obtain, for each $s' \geq 0$:

$$\begin{aligned}
&\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s'+1)}, \boldsymbol{\delta}^{(s'+1)}\right) - \mathsf{NegLog}\left(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')}\right) \\
&= \left[\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s'+1)}, \boldsymbol{\delta}^{(s'+1)}\right) - \mathsf{NegLog}\left(\boldsymbol{\mu}^{(s'+1)}, \boldsymbol{\delta}^{(s')}\right)\right] + \left[\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s'+1)}, \boldsymbol{\delta}^{(s)}\right) - \mathsf{NegLog}\left(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')}\right)\right] \\
&\leq -\frac{1}{2\gamma}\left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s')})\right\|_2^2 - \frac{\lambda_{\mathsf{lower}}^2}{2}\left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')})\right\|_1^2 \\
&\leq -\frac{R}{2}\left[\left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s')})\right\|_2^2 + \left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')})\right\|_1^2\right],
\end{aligned}$$

where the second inequality follows from the definition of R. Summing the above from 0 to any iteration $s \geq 0$, we get

$$\begin{aligned}
&\sum_{s'=0}^{s}\left[\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s'+1)}, \boldsymbol{\delta}^{(s'+1)}\right) - \mathsf{NegLog}\left(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')}\right)\right] \\
&\leq -\frac{R}{2}\sum_{s'=0}^{s}\left[\left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s')})\right\|_2^2 + \left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')})\right\|_1^2\right]
\end{aligned}$$

The left hand side of the inequality above is a telescoping sum, and it is equal to $\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s+1)}, \boldsymbol{\delta}^{(s+1)}\right) - \mathsf{NegLog}\left(\boldsymbol{\mu}^{(0)}, \boldsymbol{\delta}^{(0)}\right)$, which is bounded below by $\mathsf{NegLog}^* - \mathsf{NegLog}\left(\boldsymbol{\mu}^{(0)}, \boldsymbol{\delta}^{(0)}\right)$. We thus have

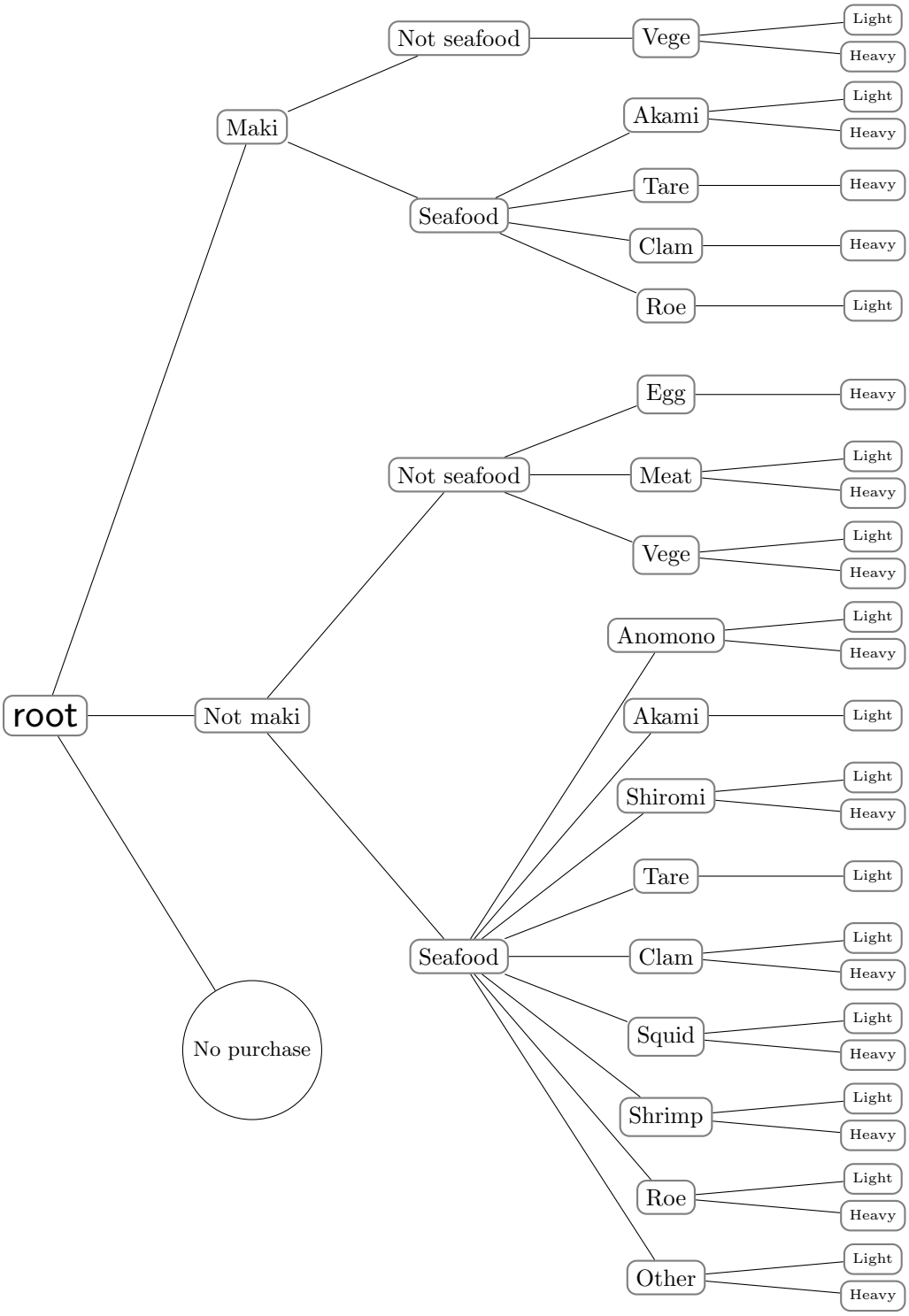
$$\mathsf{NegLog}^* - \mathsf{NegLog}\left(\boldsymbol{\mu}^{(0)}, \boldsymbol{\delta}^{(0)}\right) \leq -\frac{R}{2}\sum_{s'=0}^{s}\left[\left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s')})\right\|_2^2 + \left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')})\right\|_1^2\right] \tag{24}$$

We can now complete the proof by noting that the minimum of a collection of numbers is less than the average of that collection. Specifically,

$$\begin{aligned}
&\min_{0\leq s'\leq s}\left[\left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s')}\right)\right\|_1^2 + \left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s')})\right\|_2^2\right] \\
&\leq \frac{1}{s+1}\sum_{s'=0}^{s}\left[\left\|\nabla_{\boldsymbol{\mu}}\mathsf{NegLog}\left(\boldsymbol{\mu}^{(s')}, \boldsymbol{\delta}^{(s)}\right)\right\|_1^2 + \left\|\mathcal{G}_{H,\overline{\mathsf{Dom}_2}}(\boldsymbol{\delta}^{(s')})\right\|_2^2\right] \\
&\leq \frac{2}{R}\cdot\frac{1}{s+1}\left[\mathsf{NegLog}\left(\boldsymbol{\mu}^{(0)}, \boldsymbol{\delta}^{(0)}\right) - \mathsf{NegLog}^*\right],
\end{aligned}$$

where the final inequality follows from (24). The result of the theorem now follows. ■

Appendix H: Tree structure used in SUSHI Dataset study



The figure above depicts the tree structure employed in our real-world study on the SUSHI Preference Dataset; since there are 100 products (= the number of sushi varieties), we only show the non-leaf nodes. At the root node, the customer either decides to purchase a sushi variety or leave without a purchase. If the customer decides to make a purchase, she first chooses the *style*

of the sushi type: maki or not maki. Then, for each style, she decides to purchase a sushi variety based on whether or not it contains *seafood*. If the sushi contains seafood, she further chooses from nine minor groups: aomono (blue-skinned fish), akami (red meat fish), shiromi (white-meat fish), tare (something like baste; for eel or sea eel), clam or shell, squid or octopus, shrimp or crab, roe, and other seafood. Otherwise, she chooses from three minor groups: egg, meat other than fish, and vegetables. Lastly, she determines whether to purchase a sushi type with heavy/oily or light taste.

Appendix I: Additional Performance Measure

Here, we compare the PGD, Knitro and A-MM benchmarks on the root mean square error (RMSE) metric. For the simulation study, let $(\mathcal{S}_i^m : i = 1, 2, \ldots, 700)$ denote the sampled offer-sets for instance m. Then, we compute the average RMSE value across all the 100 instances for each algo $\in \{\text{A-MM}, \texttt{Knitro}, \text{PGD}\}$ as follows:

$$\text{RMSE}^{\text{algo}} = \frac{1}{100}\sum_{m=1}^{100}\sqrt{\frac{1}{60}\sum_{i=1}^{60}\frac{1}{|\mathcal{S}_i^m|}\sum_{\ell\in\mathcal{S}_i^m}\left(\mathbb{P}_\ell\left(\mathcal{S}_i^m;\boldsymbol{\mu}^{\text{true},m},\boldsymbol{\lambda}^{\text{true},m}\right)-\mathbb{P}_\ell\left(\mathcal{S}_i^m;\boldsymbol{\mu}^{\text{algo},m},\boldsymbol{\lambda}^{\text{algo},m}\right)\right)^2},$$

where for each instance m, $\mathbb{P}_\ell\left(\mathcal{S}_i^m;\boldsymbol{\mu}^{\text{true},m},\boldsymbol{\lambda}^{\text{true},m}\right)$ is the ground-truth purchase probability, and $\mathbb{P}_\ell\left(\mathcal{S}_i^m;\boldsymbol{\mu}^{\text{algo},m},\boldsymbol{\lambda}^{\text{algo},m}\right)$ is the estimated purchase probability by method algo, of product ℓ in offer-set $\mathcal{S}_i^m$. We report the RMSE values as well as the percentage improvements $100\times\left(\text{RMSE}^{\text{algo}}-\text{RMSE}^{\text{A-MM}}\right)/\text{RMSE}^{\text{algo}}$ for algo $\in\{\text{PGD},\texttt{Knitro}\}$ in Figure EC.1. Similar to the NegLogGap values in the main text, the A-MM method achieves significantly lower RMSE than the benchmarks for all ground-truth problem sizes except the two smallest ones.

For the Sushi dataset study, let $(\mathcal{S}_i^m : i = 1, 2, \ldots, 700)$ denote the training offer-sets for instance m, and $(\mathcal{S}_i^m : i = 701, 702, \ldots, 1000)$ denote the test offer-sets. Then, we compute the average RMSE value on the training and test offer-sets for each method algo $\in\{\text{A-MM},\text{PGD},\texttt{Knitro}\}$ as follows:

$$\text{RMSE}^{\text{algo}}_{\text{Train}} = \frac{1}{400}\sum_{m=1}^{400}\sqrt{\frac{1}{700}\sum_{i=1}^{700}\frac{1}{|\mathcal{S}_i^m|}\sum_{\ell\in\mathcal{S}_i^m}\left(\text{sales}_\ell(\mathcal{S}_i^m)-\mathbb{P}_\ell\left(\mathcal{S}_i^m;\boldsymbol{\mu}^{\text{algo},m},\boldsymbol{\lambda}^{\text{algo},m}\right)\right)^2}$$

$$\text{RMSE}^{\text{algo}}_{\text{Test}} = \frac{1}{400}\sum_{m=1}^{400}\sqrt{\frac{1}{300}\sum_{i=701}^{1000}\frac{1}{|\mathcal{S}_i^m|}\sum_{\ell\in\mathcal{S}_i^m}\left(\text{sales}_\ell(\mathcal{S}_i^m)-\mathbb{P}_\ell\left(\mathcal{S}_i^m;\boldsymbol{\mu}^{\text{algo},m},\boldsymbol{\lambda}^{\text{algo},m}\right)\right)^2}$$

where $\text{sales}_\ell(\mathcal{S}_i^m)$ is the fraction of observed sales for product ℓ in offer set $\mathcal{S}_i^m$, and $\mathbb{P}_\ell\left(\mathcal{S}_i^m;\boldsymbol{\mu}^{\text{algo},m},\boldsymbol{\lambda}^{\text{algo},m}\right)$ is the probability that a customer purchases product ℓ from the offer set $\mathcal{S}_i^m$ under the parameters estimated by algo in instance m. We report the RMSE numbers as well as the percentage improvements $100\times\left(\text{RMSE}^{\text{algo}}_{\text{Train}}-\text{RMSE}^{\text{A-MM}}_{\text{Train}}\right)/\text{RMSE}^{\text{algo}}_{\text{Train}}$ and $100\times\left(\text{RMSE}^{\text{algo}}_{\text{Test}}-\text{RMSE}^{\text{A-MM}}_{\text{Test}}\right)/\text{RMSE}^{\text{algo}}_{\text{Test}}$ for algo $\in\{\text{PGD},\texttt{Knitro}\}$ in Figure EC.2. Similar to the NegLogImp values in the main text, the A-MM method achieves lower RMSE values than the benchmarks on both training and test data, and is robust to initialization.

Degree	Height	# Prods.	# Nodes	λ_{lower}	RMSE ($\times 10^{-5}$)			RMSE Impr.		% better	
					A-MM	PGD	Knitro	over PGD	over Knitro	over PGD	over Knitro
5	4	625	781	0.50	5.02	6.52	7.21	23.0%	30.4%	72	91
				0.10	8.60	7.73	9.32	-11.2%	7.8%	40	74
				0.01	9.75	9.49	9.89	-2.8%	1.4%	50	68
	5	3,125	3,906	0.50	1.18	2.40	2.12	50.6%	44.1%	100	99
				0.10	2.15	2.87	3.45	25.1%	37.5%	95	100
				0.01	2.68	3.53	3.95	24.1%	32.2%	100	100
6	4	1,296	1,555	0.50	2.38	5.54	5.89	56.9%	59.5%	100	100
				0.10	4.32	6.45	6.95	33.0%	37.8%	90	100
				0.01	5.04	8.15	7.45	38.2%	32.4%	100	100
	5	7,776	9,331	0.50	0.46	1.57	3.19	70.8%	85.6%	100	100
				0.10	0.87	1.84	6.01	52.5%	85.5%	100	100
				0.01	1.09	2.32	6.57	53.2%	83.5%	100	100
7	4	2,401	2,801	0.50	1.26	4.42	2.99	71.6%	58.0%	100	100
				0.10	2.36	5.62	6.22	58.1%	62.1%	90	100
				0.01	2.80	6.39	6.56	56.2%	57.4%	100	100
	5	16,807	19,608	0.50	0.20	0.99	—	79.8%	—	100	—
				0.10	0.40	1.11	—	63.8%	—	100	—
				0.01	0.50	1.38	—	63.7%	—	100	—
8	4	4,096	4,681	0.50	0.85	3.59	—	76.3%	—	100	—
				0.10	1.37	4.28	—	68.0%	—	100	—
				0.01	1.63	5.07	—	67.9%	—	100	—
	5	32,768	37,449	0.50	0.10	0.62	—	83.9%	—	100	—
				0.10	0.17	0.78	—	78.2%	—	100	—
				0.01	0.23	0.91	—	74.7%	—	100	—

Figure EC.1 Comparison of the performances of PGD, `Knitro`, and our proposed A-MM method in fitting tree logit models to choice data. The columns "Degree", "Height", and λ_{lower} report the degree of each non-leaf node, the height of the tree, and the lower bound on the nest dissimilarity parameters, respectively. The columns "# Prods." and "# Nodes" report the number of products and the number of nodes in the tree, respectively. The columns under A-MM, PGD and `Knitro` report the average RMSE for each method. The columns under "RMSE Impr." reports the percentage improvement in the average RMSE value that our A-MM method achieves over the benchmarks. Finally, the columns under "% better" reports the percentage of instances in which the A-MM method obtains a lower RMSE value than the corresponding benchmark. The `Knitro` benchmark is unable to complete even a single iteration for large problem sizes, and we use "—" to denote such instances. All numbers under the "RMSE Impr." columns are significantly different from zero at the 1% significance level under a paired samples t-test.

Initialization	Train RMSE ($\times 10^{-4}$)			Test RMSE ($\times 10^{-4}$)			Impr. over PGD		Impr. over Knitro	
	A-MM	Knitro	PGD	A-MM	Knitro	PGD	Train	Test	Train	Test
0/1 start	13.64	13.72	14.35	13.69	13.78	14.40	5.23%	5.20%	0.59%	0.61%
warm start	13.64	13.67	13.79	13.69	13.73	13.85	1.16%	1.15%	0.27%	0.27%

Figure EC.2 Comparison of the performances of PGD, Knitro and our proposed A-MM method in fitting tree logit models to the Sushi Preference Dataset. The first (resp. second) row reports the performance when the methods are initialized using *0/1 start* (resp. *warm start*). The second, third and fourth columns report the RMSE value of A-MM, Knitro and PGD on the training data, while the fifth, sixth and seventh columns report the corresponding RMSE values on the test data. The eight and tenth columns report the percentage improvement in the average RMSE of the A-MM method over the PGD and Knitro benchmarks, respectively, on the training data. The corresponding improvements on the test data are reported in columns nine and eleven. All numbers under the "Impr." columns are significantly different from zero at the 1% significance level under a paired sample t-test.