Learning a Mixture of Two Multinomial Logits

Flavio Chierichetti[1] Ravi Kumar[2] Andrew Tomkins[2]

Abstract

The classical Multinomial Logit (MNL) is a behavioral model for user choice. In this model, a user is offered a slate of choices (a subset of a finite universe of n items), and selects exactly one item from the slate, each with probability proportional to its (positive) weight. Given a set of observed slates and choices, the likelihood-maximizing item weights are easy to learn at scale, and easy to interpret. However, the model fails to represent common real-world behavior. As a result, researchers in user choice often turn to mixtures of MNLs, which are known to approximate a large class of models of rational user behavior. Unfortunately, the only known algorithms for this problem have been heuristic in nature. In this paper we give the first polynomial-time algorithms for exact learning of uniform mixtures of two MNLs. Interestingly, the parameters of the model can be learned for any n by sampling the behavior of random users only on slates of sizes 2 and 3; in contrast, we show that slates of size 2 are insufficient by themselves.

1. Introduction

In this paper we study the problem of learning a uniform mixture of two multinomial logistic models from data. Our work is situated in the literature of discrete choice as follows. The most well-studied class of "rational" choice behavior is the class of Random Utility Models, introduced by Marschak (1960), and described below. Mixtures of multinomial logistic models have been widely used in discrete choice since 1980 (Boyd & Mellman, 1980; Cardell & Dunbar, 1980), and are of particular interest because they are known to ϵ-approximate any Random Utility Model (McFadden & Train, 2000). However, despite a long history of study and broad use in practice, there are no known polynomial-time algorithms to learn (exactly or approximately) any non-trivial mixture of multinomial logistic models. We give the first such result: an algorithm to learn uniform mixtures of two multinomial logistic models. We now give a little more background.

A *multinomial logistic model* (usually called an *MNL*) over a universe $\mathcal{U}$ of items provides a specific mapping from any non-empty subset $S \subseteq \mathcal{U}$ to a distribution over S. The model requires a weight function $w : \mathcal{U} \to \mathbb{R}^+$ that gives a positive weight to each item in the universe. The model then assigns probability to each $u \in S$ proportional to its weight: $\Pr[u \mid S] = w(u)/\sum_{v \in S} w(v)$.

These models are frequently employed in the setting of *discrete choice*, in which a user must select exactly one item from a set of alternatives. If $S \subseteq \mathcal{U}$ gives the alternatives, the MNL then provides a distribution representing the likelihood that each item in S will be selected. Such models are employed in many settings: selection of a piece of music, a mode of transportation, a brand of toothpaste, and so forth. We note in passing that the weight function may be generalized in many ways. Rather than mapping from u to $w(u)$, it may instead be defined in terms of features of u (allowing easy generalization to unseen objects) or in terms of features of the particular situation (for instance depending on properties of the user making the choice). In our work we do not consider such generalizations; we assume that w simply maps an item to a positive real-valued weight.

Given sufficient examples of subsets S with resulting choices of a particular $u \in S$, it is possible to estimate the weight function w using maximum likelihood estimation. The estimation is convex and is easily solved at large scale by gradient ascent methods. As a result, MNL is widely used in practice. Furthermore, in the context of the rapid changes underway in machine learning due to deep networks, it is the standard top layer of multiclass classification networks, where it goes by the name *softmax* layer.

A problematic example. The definition of MNL posits a single fixed weight function, which imposes certain restrictions on the behavior of the model across related subsets. For example, consider a distribution of authors, each of whom wishes to submit a paper to a conference. The slate of available options is $\{$ICML, CVPR$\}$, and based on the distribution, 60% of authors work in vision, and submit

[*] Equal contribution [1] Sapienza University, Rome, Italy [2] Google, Mountain View, CA. Correspondence to: Flavio Chierichetti <flavio@di.uniroma1.it>.

Proceedings of the 35th International Conference on Machine Learning, Stockholm, Sweden, PMLR 80, 2018. Copyright 2018 by the author(s).

to CVPR. Now imagine that a new ML conference called ICLR is introduced, so the new universe is {ICML, ICLR, CVPR}. Here, we expect that—on each slate containing CVPR and at least another conference—CVPR will still be preferred by roughly 60% of the authors, and that—on each slate containing both ICML and ICLR—a random author will prefer ICML with roughly the same probability of preferring ICLR. These two constraints are incompatible with an MNL. Indeed, in an MNL, for CVPR to win with probability 60% on the full slate, it must be that its weight is 60% of the total weight of the three conferences. Thus, for ICML and ICLR to be chosen with the same probability it must be that they have the same weight of 20%; but, if that is the case, then the slate {ICML, CVPR} will let CVPR win with probability 75%. This is a direct consequence of the definition of MNL: any new alternative introduced to a slate of options must decrease the likelihood of every other option by the same fraction. As in the example, this may result in undesirable restrictions in model behavior.

Mixture of MNLs. In the previous example, the issue is that theoreticians and vision researchers represent two distinct populations. Modeling the union with a single MNL results in the problem described above. On the other hand, allowing a mixture of these two populations, each represented by a population-specific MNL, will result in the correct behavior: vision researchers will employ an MNL that selects CVPR no matter what ML conferences are available, while theoreticians will employ an MNL that selects from among whatever ML conferences are present. Introducing ICLR will now cause theoreticians to move from 100% ICML to some mixture of the two ML conferences, while all the vision researchers who used to submit to CVPR will continue to do so. The mixture of two MNLs is no longer bound by the restriction that a new item in the slate must "cannibalize" equally from all other items.

In fact, moving from a single MNL to a mixture of MNLs is surprisingly powerful, as we now describe. We introduced MNL as a particular function family mapping any $S \subset \mathcal{U}$ to a distribution over S, based on the specification of a weight function. We may broaden the function family by removing the restriction that the likelihood of each item is always proportional to its fixed weight. The Random Utility Model mentioned above (Marschak, 1960) is defined, not by a weight function, but by a distribution over *value vectors*, where each value vector assigns a value to each item of $\mathcal{U}$. A user draws a value vector i.i.d. from the distribution, then behaves rationally by choosing the item of S with maximum value. The distribution over value vectors induces a distribution over any subset S. It is easy to show that any Random Utility Model may be approximated arbitrarily closely by a mixture of MNLs (McFadden & Train, 2000). Hence, the problem of learning mixtures of MNLs is equivalent to the problem of learning the large family of Random Utility Models.

For this reason, mixtures of MNLs are commonly employed in discrete choice settings. Unfortunately, the model learning is performed using heuristic techniques with no guarantees of optimality. Other than degenerate mixtures of a single MNL, to our knowledge, there are no results (positive or negative) regarding optimal learning of mixtures of MNLs, despite these models being well-studied in expressive power and commonly employed by practitioners with numerous libraries available to perform learning by heuristic approaches. We take a first step towards remedying this situation by resolving positively the question of learning uniform mixtures of two MNLs.

Our results. Let $a, b : \mathcal{U} \to \mathbb{R}^+$ be two weight functions. The uniform 2-MNL (a, b) assigns to item u in subset $S \subseteq \mathcal{U}$ the probability $\frac{1}{2} \cdot \frac{a(u)}{\sum_{v \in S} a(v)} + \frac{1}{2} \cdot \frac{b(u)}{\sum_{v \in S} b(v)}$. We show the following:

- *Uniqueness:* If $|\mathcal{U}| \geq 3$ and 2-MNLs (a, b) and (a', b') agree on every $S \subseteq \mathcal{U}$ satisfying $|S| \leq 3$, then either $a = a', b = b'$ or $a' = b, a = b'$.
- *Identifiability:* There is an algorithm that learns any 2-MNL (a, b) in time $O(|\mathcal{U}|)$.

The algorithm for identifiability builds on a reconstruction oracle derived from the uniqueness. This oracle, when presented with any slate of size at most 3, returns the distribution over items of the slate induced by the mixture. In contrast we show that slates of size 2 alone are insufficient for reconstruction and hence the oracle is optimal in terms of the slate size. If the oracle can be queried adaptively, we show an algorithm that makes $O(|\mathcal{U}|)$ queries, which we show to be optimal. For the non-adaptive case, we show an algorithm that makes $O(|\mathcal{U}|^2)$ queries, also optimal.

Establishing the uniqueness for slates of size at most 3, while seemingly a simple "finite" problem, turns out to be technically challenging. The underlying question involves studying the uniqueness of solution to a system of quartic multivariate polynomials, derived from the unknown parameters of the mixture model. Through a series of reductions and delicate case analyses, we obtain several structural properties of this polynomial system, which we use to prove uniqueness. The tools we develop for showing uniqueness could be of independent interest and might have applications in other algorithmic discrete choice settings.

Roadmap. In Section 2 we discuss related work and in Section 3 we introduce the notation. In Section 4, we prove lower bounds on the slate sizes that must be queried to allow reconstruction. In Section 5.1, we show lower bounds on the numbers of adaptive and non-adaptive queries for reconstruction and in Section 5.2 we show algorithms with matching query complexity. These algorithms are based on the uniqueness result that we mentioned above, which we prove in Section 6. In the Supplementary Material we discuss relaxations of the sampling oracle (Section A), and k-MNLs with lower bounds for their reconstruction (Section B), and the missing proofs (Section C).

2. Related Work

Multinomial logit (MNL) was initially introduced in the context of two-item slates by Bradley & Terry (1952), and was then extended to its current form by Luce (1959). However, the idea behind the formulation may be traced back to the earlier work by Zermelo (1928) in scoring of chess players. The extension from MNL to mixtures of MNL (also known as *mixed logit*) was developed in the choice literature jointly in 1980 by Boyd & Mellman (1980) and Cardell & Dunbar (1980). McFadden & Train (2000) show that mixed logit models are capable of approximating any random utility model, although the construction they apply may result in mixed logits of exponential size. Recently, Chierichetti et al. (2018) study choice models that are represented by distributions over permutations of the items in the universe; they show a series of lower bounds in that model. Train (2009) provides an overview of the body of motivations for studying mixtures of multinomial logits in the theory of discrete choice.

Learning of mixture models is well-studied in the machine learning community, dating back to early work of Pearson (1894) in a biological setting, studying the evolution of populations of crabs. Computational models for mixtures may be traced back to the classical k-means clustering algorithm (MacQueen, 1967), which represents data using a mixture of clusters, each represented by a centroid. More generally, the EM algorithm (Dempster et al., 1977) provides a heuristic to learn general forms of mixture models, with no guarantees on correctness or convergence rate. Much literature has appeared on the related problem of learning mixtures of Gaussians, under various separation assumptions; see for example the papers of Kalai et al. (2010); Moitra & Valiant (2010).

With respect to mixtures of MNLs, there are many works discussing heuristic approaches based on simulation (Train, 2009; Guevara & Ben-Akiva, 2013; Hurn et al., 2003; Ge, 2008). However, work in Computer Science is less common. Rusmevichientong et al. (2014) study a problem of selecting products to offer in order to maximize revenue over users defined by a mixture of MNLs. They show this problem is NP-hard even for mixtures of 2 MNLs. Blanchet et al. (2016) again study revenue optimization in a discrete choice setting, and present a Markov Chain-based solution that generalizes mixtures of MNLs. In fact, Oh & Shah (2014) characterizes the problem of learning a 2-MNL as "infeasible in general" given current techniques.

Recall that in our case, the oracle returns the distribution of choosing items in a given slate; as we show in Appendix A, this oracle can be well approximated from choice processes by sampling. Some work on learning mixtures of MNLs assume an oracle more powerful and less realistic than ours. For instance, Oh & Shah (2014) study the problem of learning a k-MNL using an oracle that returns the relative ordering of a number of (disjoint and/or partially overlapping) pairs of objects, as sampled from the same (random) MNL. With this oracle, one can use some of the pairs to get clues about which of the k MNLs produced a given sample, and the remaining pairs to estimate the relative weights of their elements in that specific MNL. Oh & Shah (2014) study the sample complexity of this problem in various pairs-selecting random models. Zhao et al. (2016) also study the problem of learning a k-MNL. They focus on the necessary and sufficient conditions for identifiability, but they assume that the oracle returns the probability of observing a particular permutation of the slate. This oracle is stronger than both the one in (Oh & Shah, 2014) and ours. Ammar et al. (2014) also study learning a k-MNL, but they introduce a requirement that the weights in each MNL be well-separated, with each weight larger than the previous by some multiplicative constant.

Other models that originated in machine learning and that have been the object of active theoretical investigation include LDA-like topic models, e.g., the work of Arora et al. (2012; 2013; 2016). The goal there is to approximate the topics supporting the model, given samples (i.e., documents) from the model's distribution; note that the model cannot be queried. Another difference is that, in topic reconstruction models, the algorithm usually gets more than one sample from the unknown topic (i.e., more than one word per document) whereas, in our case the algorithm gets a single sample from the unknown MNL.

3. Preliminaries

Let $[n] = \{1, \ldots, n\}$ be the universe of items. A *slate* is any non-empty subset of $[n]$. An *s-slate* is a slate of size s.

A *multinomial logit* (1-MNL, or simply, MNL) model is fully specified by a *weight function* $a : [n] \to \mathbb{R}^+$, where $\mathbb{R}^+$ is the set of positive real numbers. In this choice model, given a slate $T \subseteq [n]$, the probability that the item $i \in T$ is chosen is given by

$$D_T^a(i) = \frac{a_i}{\sum_{j \in T} a_j},$$

where for convenience we use a_i to denote $a(i)$. We can think of $D_T^a(i)$ as the probability that item i *wins* in the slate T. Clearly, without loss of generality, $\sum_{i=1}^n a_i = 1$.

A 2-MNL $\mathcal{A} = (a, b, \mu)$ consists of weight functions a, b and a mixing weight $\mu \in (0, 1)$. Given a slate $T \subseteq [n]$, $\mathcal{A}$

first chooses the weight function a with probability μ and b with probability $1-\mu$, and then behaves as the MNL corresponding to the chosen weight function. We use $D_T^{\mathcal{A}}(i)$ to denote the probability that the mixture model $\mathcal{A}$ chooses i, given the slate T. We drop the superscript when $\mathcal{A}$ is clear from the context. If $\mu = 1/2$, we call the mixture a *uniform 2-MNL* and denote it using the notation (a, b).

The goal of the learning problem is to understand the precise conditions for *identifiability* and *unique reconstruction* of the parameters of the mixture model, i.e., the weight functions and the mixing weight. We assume an oracle access to $\mathcal{A}$ that, given a slate T and an item i, returns the value $D_T^{\mathcal{A}}(i)$, i.e., the probability i wins in the slate T. The computational quantities of interest are then the number of queries to this oracle and the size of the slates queried.

4. Warmup

In this section we present a flavor of the reconstruction problem by first considering the simple 1-MNL case. For the 1-MNL case, we observe that a linear number of 2-slate queries is sufficient to uniquely identify and learn the weight function.

Observation 1. *A 1-MNL can be reconstructed using* $(n-1)$ *2-slate queries.*

Proof. For each $i \in [n-1]$, we query the MNL using the slate $\{i, n\}$ to obtain

$$D_{\{i,n\}}(n) = \frac{a_n}{a_i + a_n}.$$

This, along with $\sum_{i=1}^{n} a_i = 1$, yields a system of linear equations in a_i whose solution yields the weight function a of the 1-MNL. □

In contrast, we next show that reconstructing a 2-MNL requires queries to larger slates. Specifically, we first show that reconstructing uniform 2-MNLs needs at least 3-slate queries.

Theorem 2. *For each* $n \geq 3$*, there exist a 1-MNL* $\mathcal{A}$ *and two uniform* 2*-MNLs* $\mathcal{A}^{(1)}$ *and* $\mathcal{A}^{(2)}$*, such that* $D_T^{\mathcal{A}} = D_T^{\mathcal{A}^{(1)}} = D_T^{\mathcal{A}^{(2)}} = 1/|T|$ *for each* T *with* $|T| \leq 2$*, but there is a* T *with* $|T| = 3$ *such that* $D_T^{\mathcal{A}^{(1)}} \neq D_T^{\mathcal{A}^{(2)}}$.

Proof. Note that each item in T has the same chance of winning. Therefore, the 1-MNL with a constant weight function a satisfies $D_T^a(i) = 1/|T|$ for $i \in T$.

Given some real number $t > 1$, we define two uniform 2-MNLs $\mathcal{A}^{(1)} = (a^{(1)}, b^{(1)}), \mathcal{A}^{(2)} = (a^{(2)}, b^{(2)})$ such that $D_T^{\mathcal{A}^{(1)}}(i) = D_T^{\mathcal{A}^{(2)}}(i) = 1/|T|$ for each $|T| \leq 2$, and such that there exists a 3-slate T such that $D_T^{\mathcal{A}^{(1)}}(i) \neq D_T^{\mathcal{A}^{(2)}}(i)$. The weight functions are defined as:

- $a_1^{(1)} = a_2^{(1)} = t$, $b_1^{(1)} = b_2^{(1)} = 1/t$, and $a_i^{(1)} = b_i^{(1)} = 1$ for each $i \in \{3, \ldots, n\}$ and
- $a_1^{(2)} = b_2^{(2)} = t$, $b_1^{(2)} = a_2^{(2)} = 1/t$, and $a_i^{(2)} = b_i^{(2)} = 1$ for each $i \in \{3, \ldots, n\}$.

We begin with the statement about 2-slates. Let $T = \{i, j\}$. If $\{i, j\} \cap \{1, 2\} = \varnothing$, then both $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$ clearly induce the uniform distribution on T. If $T = \{1, 2\}$ then, since the mixture is uniform, the probability that 1 gets selected is $1/2$ with $\mathcal{A}^{(1)}$ and is $\frac{1}{2} \cdot \frac{t}{t+1/t} + \frac{1}{2} \cdot \frac{1/t}{t+1/t} = \frac{1}{2}$ with $\mathcal{A}^{(2)}$. Finally, if $T = \{1, i\}$ or $T = \{2, i\}$, for some $i \in \{3, \ldots, n\}$, then the probability of selecting i is equal, with both $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$, to $\frac{1}{2}\frac{1}{1+t} + \frac{1}{2}\frac{1}{1+1/t} = \frac{1}{2}\frac{1}{1+t} + \frac{1}{2}\frac{t}{t+1} = \frac{1}{2}$.

On the other hand, consider the 3-slate $T = \{1, 2, i\}$, for any $i \in \{3, \ldots, n\}$. In this case, we have $D_T^{\mathcal{A}^{(1)}}(i) = \frac{1}{2} \cdot \frac{1}{2t+1} + \frac{1}{2} \cdot \frac{1}{2/t+1} = \frac{1}{2} - O\left(\frac{1}{t}\right)$, while $D_T^{\mathcal{A}^{(2)}}(i) = \frac{1}{2} \cdot \frac{1}{t+1+1/t} + \frac{1}{2} \cdot \frac{1}{t+1+1/t} = O\left(\frac{1}{t}\right)$. Thus, $\lim_{t \to \infty} D_T^{\mathcal{A}^{(1)}}(i) - D_T^{\mathcal{A}^{(2)}}(i) = \frac{1}{2}$. □

We next consider non-uniform 2-MNLs and show that even 3-slates are not enough for unique reconstruction. For ease of exposition, we will use a weight function where exactly one of the weights is zero; this can be easily modified so that the construction has only positive weights.

Theorem 3. *For each* $n \geq 3$*, there exists two* 2*-MNLs* $\mathcal{A}^{(1)}$ *and* $\mathcal{A}^{(2)}$*, each with mixing weight* $2/3$ *such that* $D_T^{\mathcal{A}^{(1)}} = D_T^{\mathcal{A}^{(2)}}$ *for each* T *with* $|T| \leq 3$*, but there is a* T *with* $|T| = 4$ *such that* $D_T^{\mathcal{A}^{(1)}} \neq D_T^{\mathcal{A}^{(2)}}$.

Proof. For $i \in [2]$, let $\mathcal{A}^{(i)} = (a^{(i)}, b^{(i)}, 2/3)$. Let $a_j^{(i)} = b_j^{(i)} = 1$ for $j \in [n] \setminus \{1\}$. Let $a_1^{(1)} = 4/9$, $a_1^{(2)} = 64$ and $b_1^{(1)} = 4$, $b_1^{(1)} = 0$. For both the 2-MNLs, the mixing weight $\mu = 2/3$.

By the definition of $a^{(\cdot)}$ and $b^{(\cdot)}$, all the slates that do not contain item 1 will have their winner chosen uniformly at random, i.e., $D_T^{\mathcal{A}^{(1)}}(i) = D_T^{\mathcal{A}^{(2)}}(i) = 1/|T|$ for each $i \in T \subseteq [n] \setminus \{1\}$.

We now focus on 2- and 3-slates $T, T \ni 1$. For each $i \in [2]$, by construction, $D_T^{\mathcal{A}^{(i)}}(j) = D_T^{\mathcal{A}^{(i)}}(j')$ for any $j, j' \neq 1$. Moreover,

- if $|T| = 2$, then $D_T^{\mathcal{A}^{(1)}}(1) = \frac{2}{3} \cdot \frac{4/9}{4/9+1} + \frac{1}{3} \cdot \frac{64}{64+1} = \frac{8}{15} = \frac{2}{3} \cdot \frac{4}{4+1} + \frac{1}{3} \cdot \frac{0}{0+1} = D_T^{\mathcal{A}^{(2)}}(1)$;
- if $|T| = 3$, then $D_T^{\mathcal{A}^{(1)}}(1) = \frac{2}{3} \cdot \frac{4/9}{4/9+2} + \frac{1}{3} \cdot \frac{64}{64+2} = \frac{4}{9} = \frac{2}{3} \cdot \frac{4}{4+2} + \frac{1}{3} \cdot \frac{0}{0+2} = D_T^{\mathcal{A}^{(2)}}(1)$.

On the other hand, if $T = \{1,2,3,4\}$, then $D_T^{\mathcal{A}^{(1)}}(1) = \frac{2}{3} \cdot \frac{4/9}{4/9+3} + \frac{1}{3} \cdot \frac{64}{64+3} = \frac{840}{2077} \neq \frac{8}{21} = \frac{2}{3} \cdot \frac{4}{4+3} + \frac{1}{3} \cdot \frac{0}{0+3} = D_T^{\mathcal{A}^{(2)}}(1)$. □

In the next sections we complement these lower bounds by developing an algorithm for learning uniform 2-MNLs that uses 2-slate and 3-slate queries.

5. Learning 2-MNLs

In this section we obtain algorithms to learn 2-MNLs using only 2-slate and 3-slate queries, complementing the slate-size lower bound in Theorem 2. Our algorithms use $O(n)$ adaptive queries or $O(n^2)$ non-adaptive queries; we will also show these query bounds are optimal for any algorithm that uses constant-sized slates.

5.1. Query Lower Bounds

To illustrate our algorithms better, we first present query lower bounds for adaptive and non-adaptive algorithms. In particular we ask how many slates of bounded size must be examined to reconstruct the weights of a 2-MNL. We show that any adaptive (resp., non-adaptive) algorithm querying only slates of constant size has to perform at least $\Omega(n)$ (resp., $\Omega(n^2)$) queries to reconstruct.

Theorem 4. *Any algorithm for 2-MNL that queries using c-slates needs $\Omega(n/c)$ queries to reconstruct; the query lower bound for non-adaptive algorithms is $\Omega(n^2/c^2)$.*

Proof. Let i, j be two distinct items in $[n]$ chosen u.a.r. We will construct two different uniform 2-MNLs, $\mathcal{A}^{(i)} = (a^{(i)}, b^{(i)})$ for $i \in [2]$, as follows. Let each MNL give a weight of 1 to each item except for i and j. Let $a_i^{(1)} = a_j^{(1)} = 2$, $b_i^{(1)} = b_j^{(1)} = 1$, and $a_i^{(2)} = b_j^{(2)} = 2$, $a_j^{(2)} = b_i^{(2)} = 1$.

If an algorithm performs no query to a slate containing both items i and j, then it cannot distinguish between $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$, and is therefore unable to learn the weights of the MNLs. Indeed, for any slate $S \subseteq [n] \setminus \{i,j\}$, we have that $D_S^{\mathcal{A}^{(1)}} = D_S^{\mathcal{A}^{(2)}}$, $D_{\{i\}\cup S}^{\mathcal{A}^{(1)}} = D_{\{i\}\cup S}^{\mathcal{A}^{(2)}}$, and $D_{\{j\}\cup S}^{\mathcal{A}^{(1)}} = D_{\{j\}\cup S}^{\mathcal{A}^{(2)}}$.

Any algorithm performing queries to slates of size at most c will need to perform $\Omega(n/c)$ queries to query at least once item i with constant probability. This proves the adaptive lower bound. In the non-adaptive case, observe that each query performed by the algorithm will cover at most $\binom{c}{2}$ different pairs. Since we need the algorithm to query i and j together to distinguish between $\mathcal{A}^{(1)}$ and $\mathcal{A}^{(2)}$, and since there are $\binom{n}{2}$ many pairs of items, the algorithm will need to perform $\Omega(n^2/c^2)$ queries to succeed with constant probability. □

5.2. Adaptive and Non-Adaptive Algorithms

We now present adaptive and non-adaptive algorithms that match the above query complexity and slate lower bounds. Our algorithms are based on a reduction to the 3-item universe case that we will present in Section 6. This constant-time algorithm *uniquely* reconstructs the weights of a 2-MNL on a universe of size 3, given the winning probabilities for all subsets of sizes 2 and 3, of the 3 items, i.e., using a total of 4 queries. For the remainder of this section we will refer to this as the *3-items algorithm*.

The main idea behind the algorithms is to invoke the 3-items algorithm on chosen 3-slates and "patch" the weights returned by this algorithm to construct the weight functions a and b. However, one has to be careful given the lower bound construction in Theorem 4. For example, consider a naive algorithm that chooses the 3-slates $\{1, n-1, n\}, \{2, n-1, n\}, \ldots, \{n-2, n-1, n\}$. Note that the items $n-1$ and $n-2$ are fixed in all the slates that are queried and hence, if the two special items $\{i,j\}$ of the lower bound construction do not satisfy $\{i,j\} \cap \{n-1, n\} \neq \varnothing$, that items i and j will never be queried together. From the lower bound construction, the algorithm will fail since it will be unable to tell whether items i and j have their larger weight in the same of the two MNLs, or in different MNLs. To circumvent this, one has to get hold of a pair of items that have different behavior in the mixture and use them as "anchors" to infer the weights of the remaining items.

We first present the algorithm that uses adaptive queries.

Theorem 5. *We can reconstruct the weights of a uniform 2-MNL using $O(n)$ adaptive queries with 2- and 3-slates.*

Proof. Let M_n be an arbitrary pairing of the items in $[n] \setminus \{1\}$, if n is odd. If n is even, let $M_n = M_{n-1} \cup \{\{2, n\}\}$. Clearly, $|M_n| = \lceil n/2 \rceil$.

We run the 3-items algorithm on each of the triples $\{1, i, j\}$, for all $\{i,j\} \in M_n$. Since each call to the 3-items algorithm performs at most 4 queries, this will cost at most $(n/2) \cdot 4 + O(1) = 2n + O(1)$ queries. For a given $\{i,j\} \in M_n$, and for $x \in \{1, i, j\}$, let $a_x(\{i,j\})$ and $b_x(\{i,j\})$ be the weights of x in a and b, as returned by the 3-items algorithm when run on $\{1, i, j\}$. We abbreviate $a_x = a_x(\{i,j\}), b_x = b_x(\{i,j\})$.

Suppose the algorithm finds that for any two distinct items $s, t \in [n]$, it holds that $a_s/a_t = b_s/b_t$, i.e., all the items have the same ratio in both the components of the mixture. This means that the 2-MNL is actually a 1-MNL and hence Observation 1 completes the argument.

Otherwise, there is a triple $\{1,i,j\}$ that contains two distinct items s,t such that, wlog, $a_s/a_t > b_s/b_t$. We use this pair of items as "anchors" to infer the rest of the weights. Indeed, we run the 3-items algorithm on each triple $\{s,t,i\}$ for each $i \in [n] \setminus \{s,t\}$. For $x \in \{i,s,t\}$, let $a_x(\{i\})$ and $b_x(\{i\})$ be the weights of x in a and b, as returned by the 3-items algorithm when run on the triple $\{s,t,i\}$. This step uses at most $4n + O(1)$ queries.

We then reorder the $a(\{i\}), b(\{i\})$ weights it returns on $\{s,t,i\}$ so that $a_s(\{i\})/a_t(\{i\}) > b_s(\{i\})/b_t(\{i\})$. Then, we set $a_i = a_i(\{i\}) \cdot a_s/a_s(\{i\})$ and $b_i = b_i(\{i\}) \cdot b_s/b_s(\{i\})$. Given the guarantees of the 3-item algorithm, it is easy to see that the 2-MNL is correctly reconstructed (up to normalization).

The total number of queries used in the algorithm is at most $6n + O(1)$. □

We next present a non-adaptive algorithm. As the lower bound in Theorem 4 suggests, the difficulty arises specifically due to the possible existence of very many pairs of items having exactly the same ratio of weights in the two MNLs; this may be viewed as a form of degeneracy. We present the algorithm below.

Theorem 6. *We can reconstruct the weights of a uniform 2-MNL on n items with $O(n^2)$ non-adaptive queries with 2- and 3-slates.*

Proof. We run the 3-items algorithm on the slate $\{1,i,j\}$ for each $\{i,j\} \in \binom{[n]\setminus\{1\}}{2}$. As in the proof of Theorem 5, if there are no pairs such that $a_s/a_t \neq b_s/b_t$, then the 2-MNL is actually a 1-MNL and Observation 1 can be used to reconstruct.

Otherwise, we can identify two items s,t such that $a_s/a_t \neq b_s/b_t$. It must be that either $a_1/a_s \neq b_1/b_s$ or $a_1/a_t \neq b_1/b_t$ (indeed, if $a_1/a_s = b_1/b_s$ and $a_1/a_t = b_1/b_t$, then $a_s/a_t = b_s/b_t$). Suppose, wlog, that $a_1/a_s \neq b_1/b_s$. This allows us to obtain the weights of items 1 and s. Moreover, for each $i \in [n] \setminus \{1,s\}$, the 3-items algorithm has been run on $\{1,s,i\}$. Therefore, we can compute the weights of item i, as in the proof of Theorem 5, hence reconstructing the 2-MNL.

The non-adaptive algorithm that we described performs $4\binom{n-1}{2} = 2n^2 - O(n)$ queries. □

Note that if noise is added to the weights, or if the MNL's are otherwise guaranteed to have no equiweighted items for any triple, then the linear-time bound will also apply in the non-adaptive case.

6. Learning a 2-MNL on a 3-Item Universe

In this section we focus our attention on a universe of size 3, i.e., $n = 3$. As we saw before, our algorithms use this as a building block to work for all n.

At first glance, this problem is apparently simple, for example, there are only four unknowns in a uniform 2-MNL on $n = 3$ and there are five known free quantities (one free winning probability from each of the three 2-slates, and two free winning probabilities from the 3-slate) to possibly pin down the unknowns. However, such arguments can be deceptive and fallacious.[1] The number of unknowns and the number of available quantities do not have a simple relationship since the system, as we will see, is nonlinear. Furthermore, the *uniqueness* of the solution, given the known quantities, is not obvious and establishing it is crucial to solving the reconstruction problem. Trying to do this through automatic symbolic methods quickly runs into computational issues, as we found. This forces an analytic approach to study the multivariate polynomial system, which also yields interesting insights into the structure of the system implied by the uniform 2-MNL.

To make the exposition simpler, let the items of the universe be indexed by $\{i,j,k\}$. Again, without loss of generality, $a_i + a_j + a_k = 1 = b_i + b_j + b_k$. Note that

$$D_{\{x,y\}}(x) = \frac{1}{2}\left(\frac{a_x}{a_x + a_y} + \frac{b_x}{b_x + b_y}\right),$$
$$D_{\{x,y,z\}}(x) = \frac{a_x + b_x}{2},$$

for $\{x,y,z\} = \{i,j,k\}$.

In this section we show that the sequence of functions $\mathcal{D} = \left(D_{\{i,j,k\}}(\cdot), D_{\{i,j\}}(\cdot), D_{\{j,k\}}(\cdot), D_{\{i,k\}}(\cdot)\right)$ *uniquely* determines $a_i, a_j, a_k, b_i, b_j, b_k$ (up to reordering) and present an algorithm to find them. As we saw in Section 5, this implies we can reconstruct the 2-MNL by querying 2-slates and 3-slates. This proof of uniqueness requires a few steps that we will sketch out now as a roadmap of this section. We first introduce some key notions. We say the uniform 2-MNL $((a_i,a_j,a_k),(b_i,b_j,b_k))$ is *consistent* with $\mathcal{D}$, if it gives the same winning probabilities as $\mathcal{D}$.

Definition 7 (Equiweightedness)**.** *For a given triple $\{i,j,k\}$ such that $a_i + a_j + a_k = 1 = b_i + b_j + b_k$, and for $\ell \in \{i,j,k\}$, we say the item ℓ is* equiweighted *if $a_\ell = b_\ell$.*

[1] To appreciate how misleading the problem difficulty can be, the case of non-uniform 2-MNL is only marginally more complex (i.e., it has only one extra unknown representing the mixing weight), but we do not currently know how to solve uniqueness in this case. In fact, as we proved in Theorem 3, queries on 2-slates and 3-slates are not sufficient to reconstruct the weights of a non-uniform 2-MNL on $n = 3$ elements, regardless of the facts that those slates give 5 known quantities, and that there are exactly 5 unknowns in a non-uniform 2-MNL.

We begin by showing in Lemma 8 how to find two items of $\{i, j, k\}$ that are ordered consistently in a and b. Next, in Lemma 9, we extend this analysis to find items that are equiweighted. From here, we perform a delicate case analysis.

If all the items are equiweighted, then the 2-MNL is actually a 1-MNL, which we recover via Observation 1. If exactly one item is equiweighted (Section 6.2), we prove uniqueness in Lemma 10. We then proceed to the hardest case (Section 6.3) in which no item is equiweighted. Per Lemma 8 we may order the items $\{i, j, k\}$, and the weights a, b, such that $a_j > b_j$ and $a_k > b_k$. Lemma 11 then shows a bijection from a_j to a_k, and another bijection from a_k to a_j. The remainder of the proof proceeds by contradiction. Corollary 12 employs the form of the bijections to develop ordering constraints on the weights of items of two hypothesized distinct 2-MNLs generating the same slate probabilities. Theorem 13 then shows that the existence of two distinct 2-MNLs yields a contradiction.

In Appendix 6.4, we discuss the algorithmic implications of the uniqueness result. In particular, we show that the weights of a uniform 2-MNL on 3 elements can be found efficiently if one has access to the sequence of functions $\mathcal{D}$. Finally, in Appendix A, we show that, under a mild separation assumption on the weights, having access to a sample oracle (instead of to the exact winning probabilities in $\mathcal{D}$) is still sufficient for polynomial-time reconstruction.

6.1. Ordering and Equiweightedness of Items

We start by proving two technical statements that allow us to evaluate the relation between a_ℓ and b_ℓ, for $\ell \in \{i, j, k\}$, using just the winning probabilities on the subslates of $\{i, j, k\}$. The first result allows us to pinpoint the two items in $\{i, j, k\}$ that order their a and b weights in the same manner.

Given $x, y \in \{i, j, k\}$, $x \neq y$, define the predicate

$$P_{x,y} \triangleq [D_{\{x,y\}}(x) \cdot D_{\{x,y,z\}}(y) \geq D_{\{x,y\}}(y) \cdot D_{\{x,y,z\}}(x)].$$

Lemma 8. *$(a_x - b_x) \cdot (a_y - b_y) \geq 0$ iff $P_{x,z} \wedge P_{y,z}$, i.e., the relative ordering of a_x, b_x matches the relative ordering of a_y, b_y iff $P_{x,z} \wedge P_{y,z}$.*

The next statement characterizes equiweighted items in terms of the predicates, which will allow us to identify equiweighted items, if any.

Lemma 9. *z is equiweighted iff $P_{x,y} \wedge P_{y,x} \wedge P_{z,x} \wedge P_{z,y}$.*

Note that obtaining the weight of an equiweighted item i is trivial: indeed, if i is equiweighted, then $a_i = b_i = D_{\{i,j,k\}}(i)$. Now, if two items of $\{i, j, k\}$ are equiweighted, then all of them are equiweighted, and therefore the 2-MNL is indeed a 1-MNL and can be learned using Observation 1. In the following we consider the remaining two cases: when $\{i, j, k\}$ contains exactly one equiweighted item and when no item in $\{i, j, k\}$ is equiweighted.

6.2. Uniqueness if Exactly One Item is Equiweighted

We now show that if there is a single equiweighted item in $\{i, j, k\}$, then the uniqueness follows.

Lemma 10. *Suppose i is the only equiweighted item of $\{i, j, k\}$. Then, there is a unique 2-MNL $\mathcal{A}$ (up to reordering) that is consistent with $\mathcal{D}$.*

6.3. Uniqueness With No Equiweighted Items

We now consider the remaining case where no item in $\{i, j, k\}$ is equiweighted, i.e., $a_i \neq b_i$, $a_j \neq b_j$, and $a_k \neq b_k$. Lemma 8 can be used to find the two indices in $\{i, j, k\}$ that order the weights in the two MNLs in the same manner. We assume wlog that the two indices are j, k, i.e., we assume that $(a_j - b_j)(a_k - b_k) > 0$. Wlog, by reordering, we also assume that $a_j > b_j$ and $a_k > b_k$.

The first result in this section relates the value of a_j to the value of a_k and vice versa.

Lemma 11. *Suppose that $((a_i, a_j, a_k), (b_i, b_j, b_k))$ is consistent with $\mathcal{D}$. Suppose further that the functions in $\mathcal{D}$ satisfy $P_{j,i} \wedge P_{k,i}$, $\overline{P_{i,k} \wedge P_{j,k}}$, and $\overline{P_{i,j} \wedge P_{k,j}}$.*[2] *Then,*

$$a_j = \left(D_{\{i,j\}}(j) + \frac{D_{\{i,j\}}(j)D_{\{i,j,k\}}(i) - D_{\{i,j\}}(i)D_{\{i,j,k\}}(j)}{a_k - D_{\{i,j,k\}}(k)}\right) \cdot (1 - a_k) \;\triangleq f_j(a_k), \tag{1}$$

and

$$a_k = \left(D_{\{i,k\}}(k) + \frac{D_{\{i,k\}}(k)D_{\{i,j,k\}}(i) - D_{\{i,k\}}(i)D_{\{i,j,k\}}(k)}{a_j - D_{\{i,j,k\}}(j)}\right) \cdot (1 - a_j) \;\triangleq f_k(a_j). \tag{2}$$

Moreover, $f_j(a_k)$ is decreasing for $a_k \in (D_{\{i,j,k\}}(k), 1)$, and $f_k(a_j)$ is decreasing for $a_j \in (D_{\{i,j,k\}}(j), 1)$.

Proof. Let us consider the expression for $D_{\{i,j\}}(j)$:

$$\frac{1}{2}\frac{a_j}{a_i + a_j} + \frac{1}{2}\frac{b_j}{b_i + b_j} = D_{\{i,j\}}(j). \tag{3}$$

Using $D_{\{i,j,k\}}(j) = (a_j + b_j)/2$, (3) can be rewritten as

$$\frac{a_j(a_k - b_k)}{2} = D_{\{i,j\}}(j)(1 - a_k)(1 - b_k) - D_{\{i,j,k\}}(j)(1 - a_k). \tag{4}$$

Since $a_k \neq b_k$, we can divide (4) by $a_k - b_k$ to obtain

$$\frac{1}{2}a_j = (1 - a_k)\frac{D_{\{i,j\}}(j)(1 - b_k) - D_{\{i,j,k\}}(j)}{a_k - b_k}. \tag{5}$$

[2] By Lemmas 8 and 9, this is equivalent to requiring each compatible solution $((a_i, a_j, a_k), (b_i, b_j, b_k))$, with $a_j \geq b_j$, to satisfy $a_j > b_j$ and $a_k > b_k$.

Now, we use $2(D_{\{i,j,k\}}(k) - b_k) = 2\left(\frac{1}{2}a_k + \frac{1}{2}b_k - b_k\right) = a_k - b_k$ in (5) to obtain (1). The derivation of (2) is analogous.

For the monotonicity claim, recall that $P_{j,i} \iff [D_{\{i,j\}}(j) D_{\{i,j,k\}}(i) - D_{\{i,j\}}(i) D_{\{i,j,k\}}(j) \geq 0]$. Thus by our $P_{j,i}$ assumption, the numerator of the fraction in (1) is non-negative (see Lemma 8). For any $a_k \in (D_{\{i,j,k\}}(k), 1)$, the denominator of that fraction is positive, and it increases with a_k. Since $D_{\{i,j\}}(j)$ is also positive, $f_j(a_k)$ decreases as a_k increase. The monotonicity of $f_j(a_k)$ can be proved symmetrically. □

We then get the following consequence relating the orderings of the a_j's, b_j's, a_k's, and b_k's.

Corollary 12. *Suppose the functions in $\mathcal{D}$ satisfy $P_{j,i} \wedge P_{k,i} \wedge \overline{P_{i,k} \wedge P_{j,k}} \wedge \overline{P_{i,j} \wedge P_{k,j}}$. Then, if $((a'_i, a'_j, a'_k), (b'_i, b'_j, b'_k)) \neq ((a''_i, a''_j, a''_k), (b''_i, b''_j, b''_k))$ are weights consistent with $\mathcal{D}$ and assuming wlog that $a'_j \geq b'_j$, $a''_j \geq b''_j$, it must hold that either:*

- *$a'_j > a''_j > b''_j > b'_j$ and $a''_k > a'_k > b'_k > b''_k$, or*
- *$a''_j > a'_j > b'_j > b''_j$ and $a'_k > a''_k > b''_k > b'_k$.*

Proof. By the properties of the functions in $\mathcal{D}$, and Lemmas 8 and 9, we must have $a'_j > b'_j$, $a''_j > b''_j$, $a'_k > b'_k$ and $a''_k > b''_k$. Thus, $\max(b'_j, b''_j) < D_{\{i,j,k\}}(j) < \min(a'_j, a''_j)$.

Suppose that $a'_j = a''_j$. Then, by Lemma 11, it must be that $a'_k = f_k(a'_j)$ and $a''_k = f_k(a''_j)$, and thus $a'_k = a''_k$. By the consistency of the $D_{\{i,j,k\}}(j)$ winning probability, it must also hold that $b'_j = b''_j$ and $b'_k = b''_k$. Therefore, $((a'_i, a'_j, a'_k), (b'_i, b'_j, b'_k)) = ((a''_i, a''_j, a''_k), (b''_i, b''_j, b''_k))$, and we get a contradiction.

Otherwise we have two cases.

- If $a'_j > a''_j$, then $a'_k = f_k(a'_j) < f_k(a''_j) = a''_k$, by the decreasing property of $f_k(\cdot)$ proved in Lemma 11. Then, by the consistency of the winning probabilities $D_{\{i,j,k\}}(j)$ and $D_{\{i,j,k\}}(k)$, we must have $a'_j + b'_j = a''_j + b''_j$ and $a'_k + b'_k = a''_k + b''_k$. Thus, $b'_j < b''_j$ and $b'_k > b''_k$.
- If $a'_j < a''_j$, then symmetrically we get $a'_k = f_k(a'_j) > f_k(a''_j) = a''_k$, and $b'_j > b''_j$ and $b'_k < b''_k$. □

We are now ready to prove uniqueness in the no equiweighted items case.

Theorem 13. *Suppose the functions in $\mathcal{D}$ satisfy $P_{j,i} \wedge P_{k,i} \wedge \overline{P_{i,k} \wedge P_{j,k}} \wedge \overline{P_{i,j} \wedge P_{k,j}}$. Then, there is a unique $((a_i, a_j, a_k), (b_i, b_j, b_k))$, up to reordering, consistent with $\mathcal{D}$.*

6.4. Reconstructing the Weights

Having established the uniqueness of a 2-MNL given $\mathcal{D}$, we show how to determine its weights. First, observe that, having access to the functions in $\mathcal{D}$, we can write down a system of polynomial inequalities having the unknown weights of the 2-MNL $\mathcal{A}$ as its solution:

$$\begin{cases} \frac{a_i}{a_i+a_j} + \frac{b_i}{b_i+b_j} = 2D^{\mathcal{A}}_{\{i,j\}}(i) \\ \frac{a_i}{a_i+a_k} + \frac{b_i}{b_i+b_k} = 2D^{\mathcal{A}}_{\{i,k\}}(i) \\ \frac{a_j}{a_j+a_k} + \frac{b_j}{b_j+b_k} = 2D^{\mathcal{A}}_{\{j,k\}}(j) \\ \frac{a_i}{a_i+a_j+a_k} + \frac{b_i}{b_i+b_j+b_k} = 2D^{\mathcal{A}}_{\{i,j,k\}}(i) \\ \frac{a_j}{a_i+a_j+a_k} + \frac{b_j}{b_i+b_j+b_k} = 2D^{\mathcal{A}}_{\{i,j,k\}}(j) \\ a_i + a_j + a_k = 1 \\ b_i + b_j + b_k = 1 \\ a_i, a_j, a_k, b_i, b_j, b_k > 0 \end{cases} \tag{6}$$

For a given choice of $\mathcal{D}$, the system can be solved in constant (but very large) time using, e.g., Buchberger's algorithm (Buchberger, 1976) for computing the Gröbner Bases. Thus, we get the following.

Theorem 14. *There is a constant-time algorithm that given the $\mathcal{D}$ induced by a 2-MNL on a universe of size 3, infers the unique 2-MNL consistent with $\mathcal{D}$.*

From a practical perspective, computing the weights using the Gröbner bases of the system is computationally expensive. However, one could use Lemmas 8 and 9 to obtain, in a very efficient manner, the relative ordering and the equiweightedness of the 3 items. Then, if there are 3 equiweighted items, the solution can be computed efficiently using Observation 1. If there is exactly 1 equiweighted item, then the solution can be obtained using the proof of Lemma 10. Otherwise, there are no equiweighted items, and one could use a one-dimensional grid search suggested by the bijections of Lemma 11.

7. Conclusions

In this paper, we have proposed the first algorithm for (provably) learning exactly uniform mixtures of two multinomial logits over a universe of n items. Our algorithms run in time $O(n)$ for adaptive queries, and in time $O(n^2)$ for non-adaptive queries; we have shown that our algorithms are optimal, query-wise, in the class of algorithms that perform queries to slates of constant size.

There are significant technical challenges in extending our methods to either non-uniform mixtures or mixtures of more than two components, but parts of our proof structure do generalize. We hope that the existence of our algorithm is a first step towards finding more general provable results for this important problem.

Acknowledgements

Part of this work was done while the first author was visiting Google. The first author was supported in part by a Google Focused Research Award, by the ERC Starting Grant DMAP 680153, by the SIR Grant RBSI14Q743, and by the "Dipartimenti di Eccellenza 2018-2022" grant awarded to the Dipartimento di Informatica at Sapienza.

References

Ammar, A., Oh, S., Shah, D., and Voloch, L. F. What's your choice?: Learning the mixed multi-nomial. *SIGMETRICS Perform. Eval. Rev.*, 42(1):565–566, 2014.

Arora, S., Ge, R., and Moitra, A. Learning topic models - going beyond SVD. In *FOCS*, pp. 1–10, 2012.

Arora, S., Ge, R., Halpern, Y., Mimno, D. M., Moitra, A., Sontag, D., Wu, Y., and Zhu, M. A practical algorithm for topic modeling with provable guarantees. In *ICML*, pp. 280–288, 2013.

Arora, S., Ge, R., Koehler, F., Ma, T., and Moitra, A. Provable algorithms for inference in topic models. In *ICML*, pp. 2859–2867, 2016.

Blanchet, J., Gallego, G., and Goyal, V. A Markov chain approximation to choice modeling. *Operations Research*, 64(4):886–905, 2016.

Boyd, J. H. and Mellman, R. E. The effect of fuel economy standards on the U.S. automobile market: An hedonic demand analysis. *Transportation Research Part A: General*, 14A:367–378, 1980.

Bradley, R. A. and Terry, M. E. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324, 1952.

Buchberger, B. A theoretical basis for the reduction of polynomials to canonical forms. *SIGSAM Bull.*, 10(3): 19–29, 1976. ISSN 0163-5824. doi: 10.1145/1088216.1088219.

Cardell, N. S. and Dunbar, F. C. Measuring the societal impacts of automobile downsizing. *Transportation Research Part A: General*, 14A:423–434, 1980.

Chierichetti, F., Kumar, R., and Tomkins, A. Discrete choice, permutations, and reconstruction. In *SODA*, pp. 576–586, 2018.

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

Ge, Y. *Bayesian Inference with Mixtures of Logistic Regression: Functional Approximation, Statistical Consistency and Algorithmic Convergence*. PhD thesis, Northwestern University, Evanston, IL, USA, 2008.

Guevara, C. A. and Ben-Akiva, M. E. Sampling of alternatives in logit mixture models. *Transportation Research Part B: Methodological*, 58:185–198, 2013. ISSN 0191-2615.

Hurn, M., Justel, A., and Robert, C. P. Estimating mixtures of regressions. *Journal of Computational and Graphical Statistics*, 12(1):55–79, 2003. ISSN 10618600.

Kalai, A. T., Moitra, A., and Valiant, G. Efficiently learning mixtures of two Gaussians. In *STOC*, pp. 553–562, 2010.

Luce, R. D. *Individual Choice Behavior: A Theoretical Analysis*. Wiley & Sons, New York, 1959.

MacQueen, J. B. Some methods for classification and analysis of multivariate observations. In *Proceedings of 5th Berkeley Symposium on Mathematical Statistics and Probability. 1*, pp. 281–297. University of California Press, 1967.

Marschak, J. Binary choice constraints on random utility indications. In Arrow, K. (ed.), *Stanford Symposium on Mathematical Methods in the Social Sciences*, pp. 312–329. Stanford University Press, Stanford, CA, 1960.

McFadden, D. and Train, K. Mixed MNL models of discrete response. *Journal of Applied Econometrics*, 15: 447–470, 2000.

Moitra, A. and Valiant, G. Settling the polynomial learnability of mixtures of Gaussians. In *FOCS*, pp. 93–102, 2010.

Oh, S. and Shah, D. Learning mixed multinomial logit model from ordinal data. In *NIPS*, pp. 595–603, 2014.

Pearson, K. Contributions to the mathematical theory of evolution. *Phil. Trans. Roy. Soc. London A*, 185:71–110, 1894.

Rusmevichientong, P., Shmoys, D., Tong, C., and Topaloglu, H. Assortment optimization under the multinomial logit model with random choice parameters. *Production and Operations Management*, 23(11):2023–2039, 2014.

Train, K. E. *Discrete Choice Methods with Simulation*. Cambridge University Press, 2009.

Zermelo, E. Die berechnung der turnier-ergebnisse als ein maximumproblem der wahrscheinlichkeitsrechnung. *Mathematische Zeitschrift*, 29(1):436460, 1928.

Zhao, Z., Piech, P., and Xia, L. Learning mixtures of Plackett–Luce models. In *ICML*, pp. 2906–2914, 2016.