Learning a Mixture of Two Multinomial Logits
Supplementary Material

Flavio Chierichetti*
Sapienza University
Rome, Italy
flavio@di.uniroma1.it

Ravi Kumar
Google
Mountain View, CA
ravi.k53@gmail.com

Andrew Tomkins
Google
Mountain View, CA
atomkins@gmail.com

A Approximate Oracles

In the main body of the paper, we have assumed to have access to the exact value of $D^{\mathcal{A}}(i)$. We now discuss how Theorem 14 can be used to derive algorithmic results based on an oracle that, given a slate T, can generate samples according to $D_T^{\mathcal{A}}(\cdot)$; we call these *sample queries*. For simplicity, we will assume that all the (unknown) weights of the 2-MNL are positive integers in the range $[M]$ for some $M \geq 1$.

Our first claim is that, under the above assumption on the weights range, there exists an inverse polynomial separation between the possible values of $D_T^{\mathcal{A}}(i)$.

Lemma A.1. *Let $a, a', b, b' : [n] \to [M]$ be weight functions and let $\mathcal{A} = \left(a, b, \frac{1}{2}\right)$ and $\mathcal{A}' = \left(a', b', \frac{1}{2}\right)$. Then, for $T \subseteq U, |T| = 2, 3$, if $i \in T$, then either $D_T^{\mathcal{A}}(i) = D_T^{\mathcal{A}'}(i)$ or $\left|D_T^{\mathcal{A}}(i) - D_T^{\mathcal{A}'}(i)\right| \geq \frac{1}{162M^4}$.*

Proof. Let $A = \sum_{j \in T} a_j$, $B = \sum_{j \in T} b_j$, $A' = \sum_{j \in T} a'_j$ and $B' = \sum_{j \in T} b'_j$. Then, $A, B, A', B' \leq 3M$.

We then have, $D_T^{\mathcal{A}}(i) = \frac{a(i)}{2A} + \frac{b(i)}{2B} = \frac{a(i)B + b(i)A}{2AB}$ and $D_T^{\mathcal{A}'}(i) = \frac{a'(i)}{2A'} + \frac{b'(i)}{2B'} = \frac{a'(i)B' + b'(i)A'}{2A'B'}$. Moreover,

$$D_T^{\mathcal{A}}(i) - D_T^{\mathcal{A}'}(i) = \frac{a(i)B + b(i)A}{2AB} - \frac{a'(i)B' + b'(i)A'}{2A'B'} = \frac{a(i)A'BB' + b(i)AA'B' - a'(i)ABB' - b'(i)AA'B}{2AA'BB'}.$$

Now, if $D_T^{\mathcal{A}}(i) \neq D_T^{\mathcal{A}'}(i)$, then the numerator $a(i)A'BB' + b(i)AA'B' - a'(i)ABB' - b'(i)AA'B$ must be non-zero and since the numerator is obtained by adding, subtracting, and multiplying integers, it must evaluate to a non-zero integer. We then get $\left|D_T^{\mathcal{A}}(i) - D_T^{\mathcal{A}'}(i)\right| = 1/(2AA'BB') \geq 1/\left(162M^4\right)$. □

For a 2- or 3-slate T and for a large enough constant $c > 0$, using the sampling oracle $O(cM^8 \ln(n/\delta))$ times, we can reconstruct a value $\tilde{D}_T^{\mathcal{A}}(i)$ such that $|\tilde{D}_T^{\mathcal{A}}(i) - D_T^{\mathcal{A}}(i)| \leq \frac{1}{325M^4}$, with probability at least $1 - O\left(\delta n^{-2}\right)$. By looping through the possible values of a and b on the

*Work done in part while visiting Google. 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.

items of T, we can obtain $D_T^{\mathcal{A}}(i)$, since by Lemma A.1, it will be the one value we can obtain that is within an additive distance of $\frac{1}{325M^4}$ $\left(< \frac{1}{2} \cdot \frac{1}{162M^4}\right)$ from $\tilde{D}_T^{\mathcal{A}}(i)$. Using the algorithms in Theorem 5 and Theorem 6, and a union bound, we obtain:

Theorem A.2. *Let $a, b : [n] \to [M]$ be weight functions and let $\mathcal{A} = \left(a, b, \frac{1}{2}\right)$. Then, for each small enough $\delta > 0$, with probability at least $1 - O(\delta)$ we can reconstruct the weights a and b with $O\left(M^8 n \ln(n/\delta)\right)$ adaptive or $O\left(M^8 n^2 \ln(n/\delta)\right)$ non-adaptive sample queries to 2- and 3-slates.*

B Lower Bounds for k-MNL

A *k-MNL* is a mixture of k separate MNLs. Specifically, a k-MNL $\mathcal{A}$ is given by a set $\{a^{(1)}, \ldots, a^{(k)}\}$ of weight functions and a mixing distribution μ on $[k]$. Given a slate $T \subseteq [n]$, the mixture model first chooses an index $\ell \in [k]$ according to μ and then invokes the MNL $a^{(\ell)}$. As in 2-MNL, we only focus on uniform mixing distributions, i.e., μ is uniform on $[k]$. As before, we use $D_T^{\mathcal{A}}(i)$ to denote the probability that the mixture model $\mathcal{A}$ chooses i, given the slate T.

While large parts of our proof structure for $k = 2$ generalizes to $k > 2$, there are significant technical challenges in extending our current methods to finding an algorithm for learning uniform k-MNLs. However, we can obtain some concrete slate and query lower bounds for learning uniform k-MNLs.

We first show some generalization of Theorem 2. Specifically, we show that $(k+1)$-slate queries are necessary to learn a uniform k-MNL by showing that there are instances of 1-MNL and k-MNL that are indistinguishable to any algorithm that uses only k-slate queries.

Theorem B.1. *Let $k \geq 2$ be given, and let $p(T, i) = 1/|T|$ for each $i \in T \subseteq [k]$. Then, there is a 1-MNL $\mathcal{A}$ and an infinite family of uniform k-MNLs $\{\mathcal{A}^{(x)}\}$ such that $D_T^{\mathcal{A}^{(x)}}(i) = p(T, i)$.*

Proof. Note that the definition of $p(T, i)$ says that each item in a slate (of size at most k) has the same chance of winning. Then trivially the 1-MNL with the constant weight function a satisfies $D_T^a(i) = 1/|T|$.

For each real number $x \in (0, 1)$, we will construct a uniform k-MNL $\mathcal{A}^{(x)}$ such that $D_T^{\mathcal{A}^{(x)}}(i) = 1/|T|$. For $i \in [k]$, let

$$a_j^{(i,x)} = \begin{cases} x & \text{if } j = i, \\ \frac{1-x}{k-1} & \text{if } j \in [k] \setminus \{i\}, \end{cases}$$

and $\mathcal{A}^{(x)}$ is defined to choose uniformly across the weighting functions $a^{(1,x)}, \ldots, a^{(k,x)}$.

Now, consider any k-slate T and consider any item $i \in T$. Observe that each MNL $a^{(j,x)}$ in the mixture $\mathcal{A}^{(x)}$ when $j \notin T$, gives uniform weight to the items in T. Thus, conditioning on the MNL being chosen from the set $\{a^{(j,x)}\}_{j \notin T}$, we have that the probability that i wins is exactly $|T|^{-1}$.

On the other hand, each MNL $a^{(j,x)}$ in the mixture $\mathcal{A}^{(x)}$ when $j \in T$, will give a total weight to the items of T equal to $x + (|T| - 1) \cdot \frac{1-x}{k-1}$. Moreover, if $j \in T$, then the MNL $a^{(j,x)}$ will give to item i a weight of x if $i = j$, and a weight of $\frac{1-x}{k-1}$ otherwise. Conditioning on the MNL to be chosen in the set $\{a^{(j,x)}\}_{j \in T}$, the probability of i winning is then

$$\frac{1}{|T|} \left(1 \cdot \frac{x}{x + (|T| - 1) \cdot \frac{1-x}{k-1}} + (|T| - 1) \cdot \frac{\frac{1-x}{k-1}}{x + (|T| - 1) \cdot \frac{1-x}{k-1}} \right) = \frac{1}{|T|}.$$

Therefore, for each i in a k-slate T, it holds $D_T^{\mathcal{A}^{(x)}}(i) = 1/|T|$. ∎

Next, as in the 2-MNL case, we can show query lower bounds for adaptive and non-adaptive algorithms, generalizing Theorem 4.

Theorem B.2. *Any algorithm for k-MNL that queries using c-slates needs $\Omega(n/c)$ queries to reconstruct the k-MNL; 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 k-MNLs, $\mathcal{A} = (a^{(1)}, \ldots, a^{(k)})$ and $\mathcal{B} = (b^{(1)}, \ldots, b^{(k)})$, as follows. Let each MNL give a uniform weight of 1 to each item except for i and j. In $\mathcal{A}$ and $\mathcal{B}$, let each MNL but the first two give a weight 1 to each of the items i and j. For $\mathcal{A}$, let $a_i^{(1)} = a_j^{(1)} = 2$ and $a_i^{(2)} = a_j^{(2)} = 1$. For $\mathcal{B}$, let $b_i^{(1)} = b_j^{(2)} = 2$ and $b_j^{(1)} = b_i^{(2)} = 1$.

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

Any algorithm performing queries to slates of size at most c will need to perform at least $\Omega(n/c)$ 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 $\mathcal{A}$ and $\mathcal{B}$, and since there are $\binom{n}{2}$ many pairs of items, the algorithm will needs to perform at least $\Omega(n^2/c^2)$ to succeed with constant probability. □

We now show a strong lower bound for reconstructing the winning probabilities.

Theorem B.3. *For each $k \geq 1$, with non-adaptive queries to $O(k)$-slates, the number of queries needed to learn the winning probabilities of a 2^k-MNL on a ground set of n items is $\Omega(n^{k+1})$.*

Proof. Fix $k \geq 1$, let $K = 2^k$, let the number of items n satisfy $n \geq K + 1$. Choose $K + 1$ items uniformly at random, say the ones having indices $1 \leq i_1 < \cdots < i_{k+1} \leq n$. Moreover, choose a uniform at random bit $b \in \{0, 1\}$.

The random K-MNL is constructed as follows. For each $i \in [n] \setminus \{i_1, \ldots, i_{k+1}\}$, each MNL will give weight 1 to i. Moreover, for $0 \leq t \leq K - 1$,

- the $(t + 1)$-st MNL will assign a weight of 2 (resp., 1) to item i_j if the jth bit of t is 1 (resp., 0), for each $1 \leq j \leq k$ and,
- the $(t + 1)$-st MNL will assign a weight of 2 (resp., 1) to item i_{k+1} if the parity of b equals (resp., does not equal) the parity of the weight of the binary representation of t.

The K-MNL will choose uniformly at random among its K MNLs.

Now, observe that for any sequence of k indices out of $\{i_1, \ldots, i_{k+1}\}$, regardless of b, the projection of the 2^k MNLs on those k indices will be composed of exactly all the 2^k binary words of length k. Therefore, for each slate S of cardinality at most $k + 1$, the winning probabilities of S will be uniform regardless of b.

On the other hand, any slate containing the items $i_1, \ldots, i_{k+1}$ plus any other item, will have different winning probabilities in the two K-MNLs.

It follows that if one does not look at a slate containing all the items $\{i_1, \ldots, i_{k+1}\}$ plus any other item, one cannot learn the unknown K-MNL.

Since the indices $i_1, \ldots, i_{k+1}$ are chosen u.a.r., in a non-adaptive environment, one has to look at at least $\Omega(n^{k+1}) = \Omega(n^{1+\lg K})$ slates before being able to reconstruct the K-MNL (and/or its winning probabilities). □

C Proofs

C.1 Proof of Lemma 8

Proof. We first write a chain of predicates equivalent to $P_{x,z}$:

$$\begin{aligned}
D_{\{x,z\}}(x) \cdot D_{\{x,y,z\}}(z) - D_{\{x,z\}}(z) \cdot D_{\{x,y,z\}}(x) &\geq 0 \iff \\
\left(\frac{a_x}{1-a_y} + \frac{b_x}{1-b_y}\right)(a_z + b_z) - \left(\frac{a_z}{1-a_y} + \frac{b_z}{1-b_y}\right)(a_x + b_x) &\geq 0 \iff \\
\frac{a_x(a_z+b_z)}{1-a_y} + \frac{b_x(a_z+b_z)}{1-b_y} - \frac{a_z(a_x+b_x)}{1-a_y} - \frac{b_z(a_x+b_x)}{1-b_y} &\geq 0 \iff \\
\frac{a_x b_z - a_z b_x}{1-a_y} + \frac{b_x a_z - b_z a_x}{1-b_y} &\geq 0 \iff \\
(a_x b_z - b_x a_z) \cdot \left(\frac{1}{1-a_y} - \frac{1}{1-b_y}\right) &\geq 0 \iff \\
(a_x b_z - b_x a_z) \cdot ((1-b_y) - (1-a_y)) &\geq 0 \iff \\
(a_x b_z - b_x a_z) \cdot (a_y - b_y) &\geq 0,
\end{aligned}$$

thus $P_{x,z} \iff (a_x b_z - b_x a_z)(a_y - b_y) \geq 0$ and by symmetry $P_{y,z} \iff (a_y b_z - b_y a_z)(a_x - b_x) \geq 0$.

Now, we prove that $P_{x,z} \wedge P_{y,z} \implies (a_x - b_x) \cdot (a_y - b_y) \geq 0$. By contradiction,

- $a_x > b_x$ and $a_y < b_y \implies b_z < a_z$ for $P_{x,z}$ to hold and $b_z > a_z$ for $P_{y,z}$ to hold; and
- $a_x < b_x$ and $a_y > b_y \implies b_z > a_z$ for $P_{x,z}$ to hold and $b_z < a_z$ for $P_{y,z}$ to hold.

Then, if $P_{x,z} \wedge P_{y,z}$, either $a_x > b_x$ and $a_y > b_y$, or $a_x < b_x$ and $a_y < b_y$, or $a_x = b_x$, or $a_y = b_y$. Equivalently, $(a_x - b_x)(a_y - b_y) \geq 0$.

Now, suppose that $(a_x - b_x)(a_y - b_y) \geq 0$. We consider two cases:

- if $a_x - b_x \geq 0$ and $a_y - b_y \geq 0$, then $a_z - b_z \leq 0$, therefore both $P_{x,z}$ and $P_{y,z}$ hold;
- if $a_x - b_x \leq 0$ and $a_y - b_y \leq 0$, then $a_z - b_z \geq 0$, therefore, again, both $P_{x,z}$ and $P_{y,z}$ hold. □

C.2 Proof of Lemma 9

Proof. For simplicity, let $Q_{x,y}$ denote $P_{x,y} \wedge P_{y,x}$. In a manner analogous to the proof of Lemma 8, we can prove that $Q_{x,y} \iff [D_{\{x,y\}}(x) \cdot D_{\{x,y,z\}}(y) - D_{\{x,y\}}(y) \cdot D_{\{x,y,z\}}(x) = 0]$ and $Q_{x,y} \iff [a_x b_y = b_x a_y \vee a_z = b_z]$. Recall that $P_{z,x} \iff [(a_z b_x - b_z a_x)(a_y - b_y) \geq 0]$ and $P_{z,y} \iff [(a_z b_y - b_z a_y)(a_x - b_x) \geq 0]$. We now prove the two implications.

(i) Suppose that $Q_{x,y}, P_{z,x}, P_{y,x}$ hold but by contradiction, $a_z \neq b_z$. Then, $a_x b_y = b_x a_y \triangleq \gamma$. Summing up the two inequalities induced by $P_{z,x}$ and $P_{y,x}$, we get

$$\begin{aligned}
(a_z b_x - b_z a_x) \cdot (a_y - b_y) + (a_z b_y - b_z a_y) \cdot (a_x - b_x) &\geq 0 \iff \\
a_y a_z b_x - a_z b_x b_y - a_x a_y b_z + a_x b_y b_z + a_x a_z b_y - a_z b_x b_y - a_x a_y b_z + a_y b_x b_z &\geq 0 \iff \\
a_y b_x (a_z + b_z) + a_x b_y (a_z + b_z) - 2(a_z b_x b_y + a_x a_y b_z) &\geq 0 \iff \\
2\gamma(a_z + b_z) - 2(a_z b_x b_y + a_x a_y b_z) &\geq 0,
\end{aligned}$$

thus,

$$a_z + b_z \quad \geq \quad \frac{a_z b_x b_y}{\gamma} + \frac{a_x a_y b_z}{\gamma}. \tag{1}$$

Now, if we substitute $a_x b_y$ for the first occurrence of γ and $a_y b_x$ for the second in (1), we get

$$a_z + b_z \geq a_z \frac{b_x}{a_x} + b_z \frac{a_x}{b_x}, \tag{2}$$

while if we substitute $a_y b_x$ for the first occurrence of γ and $a_x b_y$ for the second in (1), we get

$$a_z + b_z \geq a_z \frac{b_y}{a_y} + b_z \frac{a_y}{b_y}. \tag{3}$$

We consider the following two cases.

- If $a_z > b_z$, then there must exist some $w \in \{x, y\}$ such that $b_w > a_w$ (since $a_x + a_y + a_z = 1 = b_x + b_y + b_z$). By choosing the appropriate inequality among (2) and (3), we get

$$\begin{aligned} a_z + b_z \geq \; & a_z \frac{b_w}{a_w} + b_z \frac{a_w}{b_w} \; = \; \frac{a_w}{b_w}(a_z + b_z) + \left(\frac{b_w}{a_w} - \frac{a_w}{b_w} \right) a_z \\ > \; & \frac{a_w}{b_w}(a_z + b_z) + \left(\frac{b_w}{a_w} - \frac{a_w}{b_w} \right) \frac{a_z + b_z}{2} \; = \; \frac{1}{2} \left(\frac{b_w}{a_w} + \frac{a_w}{b_w} \right) (a_z + b_z). \end{aligned}$$

- If $a_z < b_z$, then there is some $w \in \{x, y\}$ such that $b_w < a_w$. Again, choosing the appropriate inequality among (2) and (3), we get

$$\begin{aligned} a_z + b_z \geq \; & a_z \frac{b_w}{a_w} + b_z \frac{a_w}{b_w} \; = \; \frac{b_w}{a_w}(a_z + b_z) + \left(\frac{a_w}{b_w} - \frac{b_w}{a_w} \right) b_z \\ > \; & \frac{b_w}{a_w}(a_z + b_z) + \left(\frac{a_w}{b_w} - \frac{b_w}{a_w} \right) \frac{a_z + b_z}{2} \; = \; \frac{1}{2} \left(\frac{a_w}{b_w} + \frac{b_w}{a_w} \right) (a_z + b_z). \end{aligned}$$

Therefore, there is always some $w \in \{x, y\}$ such that $a_z + b_z > \frac{1}{2} \left(\frac{a_w}{b_w} + \frac{b_w}{a_w} \right) (a_z + b_z)$. However, since $a_w \neq b_w$, by the AM–GM inequality, $\frac{a_w}{b_w} + \frac{b_w}{a_w} \gneq 2$, thus obtaining the contradiction $a_z + b_z > a_z + b_z$.

(ii) Suppose that $a_z = b_z$. Then, $Q_{x,y}$ trivially holds. Consider the generic $P_{z,w}$ for $\{w, w'\} = \{x, y\}$. We have shown that $P_{z,w} \iff [(a_z b_w - b_z a_w)(a_{w'} - b_{w'}) \geq 0]$. By assumption, we have $a_z = b_z$, therefore $P_{z,w} \iff [a_z(b_w - a_w)(a_{w'} - b_{w'}) \geq 0]$. Observe that if $b_w > a_w$ it must hold that $b_{w'} < a_{w'}$ (resp., if $b_w < a_w$ then $b_{w'} > a_{w'}$). Thus $a_z(b_w - a_w)(a_{w'} - b_{w'}) \geq 0$ and $P_{z,w}$ holds. □

C.3 Proof of Lemma 10

Proof. We proceed by contradiction. Assume that there exist two distinct 2-MNLs $((a'_i, a'_j, a'_k), (b'_i, b'_j, b'_k)) \neq ((a''_i, a''_j, a''_k), (b''_i, b''_j, b''_k))$ that are both consistent with the functions in $\mathcal{D}$. We show that they will be "flipped", i.e., $((a'_i, a'_j, a'_k), (b'_i, b'_j, b'_k)) = ((b''_i, b''_j, b''_k), (a''_i, a''_j, a''_k))$.

By assumptions we have that $a'_j \neq b'_j$, $a'_k \neq b'_k$, $a''_j \neq b''_j$ and $a''_k \neq b''_k$. Moreover, by Lemma 9, we have that $a'_i = b'_i = a''_i = b''_i = D_{\{i,j,k\}}(i) \triangleq x$.

Each of the two sets of weights must give the same probability to the event i wins in the slate $\{i,j\}$, i.e.,

$$\frac{1}{2}\cdot\frac{a'_i}{a'_i+a'_j}+\frac{1}{2}\cdot\frac{b'_i}{b'_i+b'_j}=D_{\{i,j\}}(i)=\frac{1}{2}\cdot\frac{a''_i}{a''_i+a''_j}+\frac{1}{2}\cdot\frac{b''_i}{b''_i+b''_j}.$$

Using the definition of x, this yields the cubic equation

$$(a'_j+b'_j-a''_j-b''_j)x^3+2(a'_jb'_j-a''_jb''_j)x^2+(a'_ja''_jb'_j+a'_jb'_jb''_j-a'_ja''_jb''_j-a''_jb'_jb''_j)x=0. \quad (4)$$

Now since $\frac{1}{2}(a'_j+b'_j)=D_{\{i,j,k\}}(j)=\frac{1}{2}(a''_j+b''_j)$, we have that $a'_j+b'_j-a''_j-b''_j=0$; we can thus drop the highest-degree term of (4). Moreover, by our boundary conditions, we can assume $0<D_{\{i,j,k\}}(i)=x<1$ and thus we can drop the $x=0$ solution as well. After these, (4) becomes

$$2(a'_jb'_j-a''_jb''_j)\cdot x+a'_jb'_j(a''_j+b''_j)-a''_jb''_j(a'_j+b'_j)=0. \quad (5)$$

Once again we use $a'_j+b'_j=a''_j+b''_j=2D_{\{i,j,k\}}(j)$ to simplify (5) to

$$(a'_jb'_j-a''_jb''_j)\cdot x+(a'_jb'_j-a''_jb''_j)D_{\{i,j,k\}}(j)=0. \quad (6)$$

Now, for (6) to be satisfied, we must either have $x=-D_{\{i,j,k\}}(j)<0$ or $a'_jb'_j=a''_jb''_j$. The former is impossible since $x=a'_i>0$. Therefore we consider the latter, i.e., $a'_j=\frac{a''_jb''_j}{b'_j}$ and apply $a'_j+b'_j=a''_j+b''_j$ again to (6), to get

$$a''_j\cdot\frac{b''_j-b'_j}{b'_j}=b''_j-b'_j. \quad (7)$$

Examining (7), if $b''_j=b'_j$, it must also hold $a''_j=a'_j$. However, since $a'_i=b'_i=a''_i=b''_i$, it must also be that $a'_k=a''_k$ and $b'_k=b''_k$, i.e., we get the desired contradiction $((a'_i,a'_j,a'_k),(b'_i,b'_j,b'_k))=((a''_i,a''_j,a''_k),(b''_i,b''_j,b''_k))$. On the other hand, if $b''_j\neq b'_j$, we can divide (7) by $b''_j-b'_j$ to get $a''_j=b'_j$. This implies, by $a'_j+b'_j=a''_j+b''_j$, that $a'_j=b''_j$. But, if $a''_i+a''_j=x+a''_j=x+b'_j=b'_i+b'_j$, it must be that $a''_k=b'_k$. Therefore, if $((a'_i,a'_j,a'_k),(b'_i,b'_j,b'_k))\neq((a''_i,a''_j,a''_k),(b''_i,b''_j,b''_k))$, it must hold that $((a'_i,a'_j,a'_k),(b'_i,b'_j,b'_k))=((b''_i,b''_j,b''_k),(a''_i,a''_j,a''_k))$, which is again a contradiction. □

C.4 Proof of Theorem 13

Proof. The assumptions on $\mathcal{D}$ guarantee, wlog, that in any (properly reordered) pair of weights $((a_i,a_j,a_k),\ (b_i,b_j,b_k))$, it holds $a_j>b_j, a_k>b_k$. We now create a system with (1) and (2) of Lemma 11:

$$\begin{cases} a_j = (1-a_k)\left(D_{\{i,j\}}(j)+\frac{D_{\{i,j\}}(j)D_{\{i,j,k\}}(i)-D_{\{i,j\}}(i)D_{\{i,j,k\}}(j)}{a_k-D_{\{i,j,k\}}(k)}\right) \\ a_k = (1-a_j)\left(D_{\{i,k\}}(k)+\frac{D_{\{i,k\}}(k)D_{\{i,j,k\}}(i)-D_{\{i,k\}}(i)D_{\{i,j,k\}}(k)}{a_j-D_{\{i,j,k\}}(j)}\right). \end{cases} \quad (8)$$

We will show that (8) has a unique solution.

For simplicity of exposition, we rewrite (8) using $t_j=D_{\{i,j,k\}}(j)$, $t_k=D_{\{i,j,k\}}(k)$, $d_j=D_{\{i,j\}}(j)$, and $d_k=D_{\{i,k\}}(k)$ to obtain

$$\begin{cases} 2d_j\cdot a_kb_k+a_jb_k+a_kb_j = 4d_jt_k-2d_j+2t_j \\ 2d_k\cdot a_jb_j+a_jb_k+a_kb_j = 4d_kt_j-2d_k+2t_k, \end{cases}$$

where $b_j = 2t_j - a_j$ and $b_k = 2t_k - a_k$. Now, suppose by contradiction that there exist two distinct solutions (a'_j, a'_k) and (a''_j, a''_k). Then, the following system in the variable x must have, as solutions, $x = 0$ and $x = 1$:

$$\begin{cases} 2d_j \cdot (x(a''_k - a'_k) + a'_k)(x(b''_k - b'_k) + b'_k) \\ +(x(a''_j - a'_j) + a'_j)(x(b''_k - b'_k) + b'_k) + (x(a''_k - a'_k) + a'_k)(x(b''_j - b'_j) + b'_j) = 4d_j t_k - 2d_j + 2t_j \\ 2d_k \cdot (x(a''_j - a'_j) + a'_j)(x(b''_j - b'_j) + b'_j) \\ +(x(a''_j - a'_j) + a'_j)(x(b''_k - b'_k) + b'_k) + (x(a''_k - a'_k) + a'_k)(x(b''_j - b'_j) + b'_j) = 4d_k t_j - 2d_k + 2t_k \\ 0 \leq x \leq 1 \end{cases}$$

By assumption we know that the system is feasible at $x = 0$ and at $x = 1$. We collect the x's:

$$\begin{cases} x^2 \cdot (2d_j(a''_k - a'_k)(b''_k - b'_k) + (a''_j - a'_j)(b''_k - b'_k) + (a''_k - a'_k)(b''_j - b'_j)) \\ +x \cdot (2d_j a'_k(b''_k - b'_k) + 2d_j(a''_k - a'_k)b'_k + (b''_k - b'_k)a'_j + (a''_j - a'_j)b'_k + (a''_k - a'_k)b'_j + (b''_j - b'_j)a'_k) \\ -(4d_j t_k - 2d_j + 2t_j) \\ = 0 \\ x^2 \cdot (2d_k(a''_j - a'_j)(b''_j - b'_j) + (a''_k - a'_k)(b''_j - b'_j) + (a''_j - a'_j)(b''_k - b'_k)) \\ +x \cdot (2d_k a'_j(b''_j - b'_j) + 2d_k(a''_j - a'_j)b'_j + (b''_j - b'_j)a'_k + (a''_k - a'_k)b'_j + (a''_j - a'_j)b'_k + (b''_k - b'_k)a'_j) \\ -(4d_k t_j - 2d_k + 2t_k) \\ = 0 \\ 0 \leq x \leq 1 \end{cases}$$

Both the quadratics need to have $\{0, 1\}$ as their set of solutions. Therefore, the two quadratics have to have their axis of symmetry at $x = \frac{1}{2}$. In other words, the derivatives of the two quadratics have to evaluate to 0 at $x = \frac{1}{2}$. We take the derivatives of the two quadratics, to get two linear equations:

$$\begin{cases} 2x \cdot (2d_j(a''_k - a'_k)(b''_k - b'_k) + (a''_j - a'_j)(b''_k - b'_k) + (a''_k - a'_k)(b''_j - b'_j)) \\ +(2d_j a'_k(b''_k - b'_k) + 2d_j(a''_k - a'_k)b'_k + (b''_k - b'_k)a'_j + (a''_j - a'_j)b'_k + (a''_k - a'_k)b'_j + (b''_j - b'_j)a'_k) = 0 \\ 2x \cdot (2d_k(a''_j - a'_j)(b''_j - b'_j) + (a''_k - a'_k)(b''_j - b'_j) + (a''_j - a'_j)(b''_k - b'_k)) \\ +(2d_k a'_j(b''_j - b'_j) + 2d_k(a''_j - a'_j)b'_j + (b''_j - b'_j)a'_k + (a''_k - a'_k)b'_j + (a''_j - a'_j)b'_k + (b''_k - b'_k)a'_j) = 0. \end{cases}$$

Since the axes of symmetry of the two quadratics were both at $x = \frac{1}{2}$, substituting $\frac{1}{2}$ for x in the two derivatives should guarantee feasibility:

$$\begin{cases} (2d_j(a''_k - a'_k)(b''_k - b'_k) + (a''_j - a'_j)(b''_k - b'_k) + (a''_k - a'_k)(b''_j - b'_j)) \\ +(2d_j a'_k(b''_k - b'_k) + 2d_j(a''_k - a'_k)b'_k + (b''_k - b'_k)a'_j + (a''_j - a'_j)b'_k + (a''_k - a'_k)b'_j + (b''_j - b'_j)a'_k) = 0 \\ (2d_k(a''_j - a'_j)(b''_j - b'_j) + (a''_k - a'_k)(b''_j - b'_j) + (a''_j - a'_j)(b''_k - b'_k)) \\ +(2d_k a'_j(b''_j - b'_j) + 2d_k(a''_j - a'_j)b'_j + (b''_j - b'_j)a'_k + (a''_k - a'_k)b'_j + (a''_j - a'_j)b'_k + (b''_k - b'_k)a'_j) = 0. \end{cases}$$

Simplifying, we get

$$\begin{cases} 2d_j(a''_k b''_k - a'_k b'_k) = a'_j b'_k + a'_k b'_j - a''_j b''_k - a''_k b''_j \\ 2d_k(a''_j b''_j - a'_j b'_j) = a'_j b'_k + a'_k b'_j - a''_j b''_k - a''_k b''_j, \end{cases}$$

and thus:

$$\begin{cases} d_j = \frac{1}{2} \frac{a'_j b'_k + a'_k b'_j - a''_j b''_k - a''_k b''_j}{a''_k b''_k - a'_k b'_k} \\ d_k = \frac{1}{2} \frac{a'_j b'_k + a'_k b'_j - a''_j b''_k - a''_k b''_j}{a''_j b''_j - a'_j b'_j}. \end{cases}$$

Observe that the numerators on the RHSs are equal to $y = a'_j b'_k + a'_k b'_j - a''_j b''_k - a''_k b''_j$. Let us consider the two denominators of the RHSs: $a''_k b''_k - a'_k b'_k$ and $a''_j b''_j - a'_j b'_j$. We will use a form of the well-known geometric principle that relates the areas of rectangles with the same perimeter. (In other words, we are going to apply a simple form of the AM-GM inequality.)

Lemma C.1. *For $x', y', x'', y'' \geq 0$, if $x' + y' = x'' + y''$ and $|x'' - y''| > |x' - y'|$, then $x'y' > x''y''$.*

Proof. Observe that $xy = ((x+y)^2 - (x-y)^2)/4$ and $x'y' = ((x'+y')^2 - (x'-y')^2)/4 = ((x+y)^2 - (x'-y')^2)/4$. Thus,

$$xy - x'y' = \frac{(x'-y')^2 - (x-y)^2}{4} = \frac{|x'-y'|^2 - |x-y|^2}{4} > 0. \quad \square$$

Recall that, we have $a'_j + b'_j = a''_j + b''_j$ and $a'_k + b'_k = a''_k + b''_k$. By the assumptions and by Corollary 12, we can assume that one of the following two alternatives holds:

- If $a'_j > a''_j > b''_j > b'_j$ and $a''_k > a'_k > b'_k > b''_k$, then by Lemma C.1, we must have $a''_j b''_j - a'_j b'_j > 0$, and $a'_k b'_k - a''_k b''_k > 0$. Thus, if the numerator y is positive then $d_j < 0$, if y is negative, then $d_k < 0$, and if $y = 0$ then $d_j = d_k = 0$. Since both d_j and d_k have to be positive, we have reached a contradiction.

- If $a''_j > a'_j > b'_j > b''_j$ and $a'_k > a''_k > b''_k > b'_k$, then $a'_j b'_j - a''_j b''_j > 0$ and $a''_k b''_k - a'_k b'_k > 0$. Thus, if $y > 0$ then $d_k < 0$, if $y < 0$ then $d_j < 0$, and if $y = 0$ then $d_j = d_k = 0$, again, a contradiction.

Hence, two distinct solutions $((a'_i, a'_j, a'_k), (b'_i, b'_j, b'_k)) \neq ((a''_i, a''_j, a''_k), (b''_i, b''_j, b''_k))$ cannot exist. $\square$