Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size: An Analysis

Manav Ramprasad

1 Introduction

This report covers Ran Raz's 2003 paper *Multi-Linear Formulas for Permanent and Determinant are of Super-Polynomial Size* and will generally follow the flow of my presentation, including some connections to my personal research interests at the end. I want to start by giving some important background, which I also covered in the beginning of my presentation. The cornerstone of algebraic complexity theory is the effort to distinguish computational efficiency from inherent intractability, formally encapsulated by the P versus NP problem's algebraic analog: VP versus VNP. Defined by Leslie Valiant in 1979, the class VP represents polynomial families that can be computed efficiently by polynomial-size arithmetic circuits. VNP represents families that can be defined efficiently (typically as exponentially large sums) but are believed to require exponential resources to compute.

The 'champions' of these two classes are the determinant and the permanent, respectively. For an $n \times n$ matrix X, the two polynomials share nearly identical formal definitions, differing only in the sign of permutations:

$$\mathrm{Det}(X) = \sum_{\sigma \in S_n} \mathrm{sgn}(\sigma) \prod_{i=1}^{n} X_{i,\sigma(i)}$$

$$\mathrm{Perm}(X) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} X_{i,\sigma(i)}$$

The alternating signs of the determinant allow for massive algebraic cancellations, enabling polynomial-time algorithms like Gaussian elimination, placing the determinant firmly in VP. The permanent lacks this property; without intermediate cancellations, computational models are forced to bear the weight of $n!$ terms, making the permanent VNP-complete. The ultimate goal of algebraic complexity is to prove VP $\neq$ VNP by demonstrating that VNP-complete polynomials cannot be efficiently computed by polynomial-size circuits.

2 Multi-Linear Formulas

Proving lower bounds against general arithmetic formulas is notoriously difficult because general formulas can leverage intermediate higher powers that eventually cancel out, acting as computational shortcuts. To bypass this, Raz restricts the computational model to *multi-linear formulas*.

An arithmetic formula Φ is multi-linear if the polynomial computed by every sub-formula contains no variables with a degree greater than 1. Raz utilizes a strict structural definition known as *syntactic multi-linearity*, which any multi-linear formula can be converted to.

Definition of Syntactic Multi-Linearity: For any product gate v with children v_1 and v_2, the sets of variables appearing in the sub-formulas of the children must be strictly disjoint: $X_{v_1} \cap X_{v_2} = \emptyset$.

By enforcing this restriction, the formula is strictly forbidden from using intermediate higher powers, thereby crippling its ability to execute efficient cancellation shortcuts. Raz's theorem proves that any multi-linear formula computing the determinant or permanent of an $n \times n$ matrix must have a super-polynomial size of $n^{\Omega(\log n)}$. I will show all of the most important results in the following sections.

3 How Raz Uses Matrix Rank

To measure the complexity of polynomials, Raz employs the partial derivatives method pioneered by Nisan and Wigderson. Let the input variables be partitioned into two disjoint sets, Y and Z, of size m. For any polynomial f, we construct a combinatorial coefficients matrix $M_f \in \mathbb{F}^{2^m \times 2^m}$. The rows represent all multi-linear monomials in Y, the columns represent monomials in Z, and the entries are the exact coefficients of the combined monomials in f.

The core complexity measure is the rank of this matrix. Rank effectively serves as a measure of "computational bandwidth." Due to the syntactic disjointness rule, the matrix of a product gate v is exactly the tensor product of its children's matrices:

$$M_v = M_{v_1} \otimes M_{v_2} \tag{1}$$

Consequently, the rank multiplies: $\text{Rank}(M_v) = \text{Rank}(M_{v_1}) \cdot \text{Rank}(M_{v_2})$. A small formula physically lacks the gates required to generate a massive matrix rank.

4 Bottlenecks and k-Weakness

We track the flow of variables through the formula tree. Let Y_v and Z_v be the variables reaching node v. We define $b(v)$ as their average and $a(v)$ as their minimum. The disparity is $d(v) = b(v) - a(v)$.

A node is considered *strictly k-unbalanced* if this disparity is large: $d(v) \geq k$. A node acts as a computational bottleneck if it heavily skews toward Y or Z. Raz defines an absolute structural failure state called *k-weakness*. A formula is k-weak if every central path from a leaf to the root encounters at least one k-unbalanced node.

According to **Lemma 4.1**, the maximum possible rank of a k-weak formula is exponentially decayed by the parameter k:

$$\text{Rank}(M_v) \leq |\Phi| \cdot 2^{b(v)-k/2} \tag{2}$$

5 The Main Result: Lemma 5.1

To force the contradiction, we establish a probabilistic trap. We assume the existence of a "small" formula Φ computing the target polynomial, bounded by $|\Phi| \leq n^{\epsilon \log n}$. We then hit this formula with a highly chaotic, randomized mapping called Assignment A.

We set a parameter $m = \lceil n^{1/3} \rceil$ and randomly select m pairs of rows and columns to form isolated 2×2 coordinate blocks. Within these blocks, we randomly inject our Y and Z variables diagonally. All other non-intersecting variables are matched to 1 or 0.

Lemma 5.1: Because the formula is assumed to be small, it lacks the structural complexity to remain balanced under this random assignment. At $\Omega(\log n)$ independent checkpoints along any central path, the assignment acts as a massive series of coin flips. By Chernoff bounds, the probability of remaining balanced at a single checkpoint is a mere $O(n^{-1/32})$. The probability that an entire path survives without hitting a k-unbalanced node decays exponentially to $n^{-\Omega(\log n)}$.

Applying a Union Bound across the small number of paths reveals that, with an overwhelming probability of $1 - o(1)$, the assignment completely unbalances the central paths. The small formula is mathematically forced into a k-weak state (where $k = n^{1/32}$).

6 Proof By Contradiction

With the formula forced into a k-weak state, we evaluate the contradiction. **1.** Under Assignment A, the formula Φ_A is k-weak. By Lemma 4.1, its maximum possible rank is strictly less than 2^m. **2.** When Assignment A is applied to the true Permanent or Determinant, the target function collapses into a specific polynomial: exactly $\prod_{i=1}^{m}(y_i \pm z_i)$. **3.** The algebraic expansion of this product forces a choice between Y and Z variables. Every Y monomial pairs with exactly one unique Z monomial. The resulting partial-derivatives matrix forms a strict permutation matrix, which holds an exact rank of 2^m.

A structural rank strictly less than 2^m cannot compute a function demanding an exact rank of 2^m. Thus, the initial assumption is false and so **multi-linear formulas for the**

permanent and determinant must be of super-polynomial size.

7 Connections to My Research Interests

The most interesting takeaway for me personally is that the separation of VP and VNP is not an abstract idea; it actually maps the precise boundaries of physical simulation, paving the way for the pursuit of quantum advantage and modern quantum chemistry algorithms.

7.1 Fermions and Determinants

In quantum chemistry, the Pauli exclusion principle dictates that fermionic wavefunctions must be perfectly antisymmetric, classically represented by Slater Determinants. Because the general determinant belongs to VP, classical machines can efficiently evaluate a single Slater Determinant in polynomial time, enabling mean-field theories like Hartree-Fock.

However, actually capturing full electron correlation in order to achieve true chemical accuracy requires computing an exponentially large linear combination of these determinants (Full Configuration Interaction). This explosion shifts the computational reality from a tractable VP problem into the intractable VNP space. Algorithms such as basis pursuit and determinant optimization (like the NOMAGIC algorithm I am currently looking into) are designed to combat this exponential wall. By aggressively optimizing the basis set and rotating the orbital basis into a highly efficient subspace, these algorithms attempt to compress the highly correlated VNP-hard reality into a restricted number of VP-computable determinants.

Furthermore, the syntactic multi-linearity constraint mirrors the physical reality of fermions. Fermionic creation operators strictly obey anti-commutation relations ($c_i^\dagger c_i^\dagger = 0$), forbidding a state from being occupied twice. Raz's result proves that classical formulas bound strictly by "no-repeating" rules (without utilizing general non-multilinear shortcuts) suffer a super-polynomial collapse in efficiency, reflecting the extreme difficulty of classically simulating highly entangled fermionic systems using strictly defined classical ansatzes.

7.2 Bosons and Permanents

While determinants govern fermions, permanents dictate the physics of identical bosons (such as photons). The permanent calculates the exact transition amplitudes of non-interacting bosons in linear optical networks. Because computing the permanent is VNP-hard, evaluating these amplitudes classically requires exponential resources. This hardness is the reason we know methods like Boson Sampling can sample from distributions that classical formulas are mathematically barred from accessing efficiently - a quantum advantage!