The Computational Complexity Column

by

V. Arvind

Institute of Mathematical Sciences, CIT Campus, Taramani
Chennai 600113, India
arvind@imsc.res.in
http://www.imsc.res.in/~arvind

The matrix scaling problem is a natural computational problem: given a matrix A can we "scale" its rows and columns so that the scaled matrix is, say, doubly stochastic. More generally, we could stipluate that the scaled matrix has some prescribed row sums and column sums. The matrix scaling problem and its generalizations have found diverse applications in algorithms and complexity in recent years. For instance, it has lead to new deterministic approximation algorithms for the permanent.

This timely and excellent survey article by Ankit Garg and Rafael Oliveira invites the reader to the fascinating research landscape of matrix scaling with its wide range of applications.

Recent progress on scaling algorithms and applications

Ankit Garg
Microsoft Research
garga@microsoft.com

Rafael Oliveira
University of Toronto
rafael@cs.toronto.edu

Abstract

Scaling problems have a rich and diverse history, and thereby have found numerous applications in several fields of science and engineering. For instance, the matrix scaling problem has had applications ranging from theoretical computer science to telephone forecasting, economics, statistics, optimization, among many other fields. Recently, a generalization of matrix scaling known as operator scaling has found applications in non-commutative algebra, invariant theory, combinatorics and algebraic complexity; and a further generalization (tensor scaling) has found more applications in quantum information theory, geometric complexity theory and invariant theory.

In this survey, we will describe in detail the scaling problems mentioned above, showing how alternating minimization algorithms naturally arise in this setting, and we shall present a general framework to rigorously analyze such algorithms. As this framework makes extensive use of concepts from invariant theory, we also provide a very gentle introduction to basic concepts of invariant theory and how they are used to analyze alternating minimization algorithms for the scaling problems. This survey is intended for a general computer science audience, and the only background required is basic knowledge of calculus and linear algebra, thereby making it accessible to graduate students and even to advanced undergraduates.

Contents

1 Introduction **1**

2 Scaling: problems and algorithms **1**
2.1 Matrix scaling 1
2.2 Operator scaling 3
2.3 Tensor scaling 5

3 Source of scaling **8**
3.1 Invariant theory: source of scaling 8
3.2 Geometric invariant theory: non-commutative duality 10
3.2.1 Hilbert-Mumford criterion 10
3.2.2 Kempf-Ness theorem 11
3.2.3 Commutative group actions: Farkas' lemma 13
3.3 Hilbert-Mumford, Kempf-Ness and scaling 13
3.3.1 Matrix scaling 14
3.3.2 Operator scaling 15
3.3.3 Tensor scaling 15

4 Analysis of scaling algorithms **16**
4.1 Potential functions for matrix scaling 18
4.2 Potential functions for operator scaling 18
4.3 Potential functions for tensor scaling 19

5 Applications of scaling **19**
5.1 Matrix Scaling 20
5.2 Operator scaling 22
5.3 Brascamp-Lieb inequalities and polytopes 23
5.4 Tensor scaling 26

6 Conclusion and open problems **27**

7 Other recent scaling works **27**

References **28**

1 Introduction

Scaling problems have been in the background of many important developments in theoretical computer science, often times in an implicit or disguised way. For instance, Forster's lower bound on the sign-rank of a matrix [For02], Linial et al.'s deterministic approximation of the permanent [LSW98] and quantitative versions of the Sylvester-Gallai theorem [BDYW11] are results which can be seen as particular instances or consequences of scaling problems

Outside of computer science, scaling algorithms have appeared (implicitly and explicitly) in many different areas, such as economics [Sto62], statistics [Sin64], optimization [RS89], telephone forecasting [Kru37], non-commutative algebra [GGOW16], functional analysis [GGOW17], quantum information theory [Gur04a] and many others.

When trying to solve a scaling problem, a natural alternating minimization algorithm comes to mind, and as such these algorithms have been proposed independently by many researchers. The analysis of such alternating minimization algorithms, on the other hand, has been a difficult task, with many different approaches being proposed for each scaling problem, and before recent works, without a unified way of analyzing such scaling algorithms. In this survey, we exposit a unified way of analyzing the natural alternating minimization algorithms, which is based on the series of works [LSW98, Gur04a, GGOW16, BGO$^+$18, BFG$^+$18].

This survey is divided as follows: in Section 2, we formally describe the three scaling problems that we study, together with the natural alternating minimization algorithms proposed for them. In Section 3 we give an elementary introduction to invariant theory, with many examples, discussing how the scaling problems defined in Section 2 are particular instances of more general invariant theory questions. In Section 4, we provide a unified, 3-step analysis of the alternating minimization algorithms proposed in Section 2, showing how invariant theory is used in the analysis of such algorithms. In Section 5, we give a detailed discussion of some of the numerous applications of scaling algorithms, providing more references for the interested reader. In Section 6 we conclude this survey, presenting further directions and future work to be done in the area, and in Section 7 we discuss related works (old and new) which we could not cover in this survey due to space and time constraints, but otherwise would perfectly fit within the scope of the survey.

2 Scaling: problems and algorithms

We first describe the various scaling problems and natural algorithms for them based on alternating minimization. Section 2.1 studies matrix scaling, Section 2.2 studies operator scaling and Section 2.3 studies tensor scaling.

2.1 Matrix scaling

The simplest scaling problem is matrix scaling, which dates back to Kruithoff [Kru37] in telephone forecasting and Sinkhorn [Sin64] in statistics. There is a huge body of literature on this problem (see for instance [RS89, KK96, LSW98, GY98, Ide16, ALOW17, CMTV17]

and references therein). In this subsection we will describe Sinkhorn's algorithm and the analysis of it done in [LSW98]. We refer the reader to the last two references above for more sophisticated algorithms for matrix scaling.

We start with a few definitions, the first being the definition of a scaling of a matrix.

Definition 2.1 (Scaling of a matrix)**.** *Suppose we are given an $n \times n$ non-negative (real) matrix A. We say that A' is a* scaling *of A if it can be obtained by multiplying the rows and columns of A by positive scalars. In other words, A' is a scaling of A if there exist diagonal matrices B, C (with positive entries) s.t. $A' = BAC$.*

Next is the definition of a doubly stochastic matrix.

Definition 2.2 (Doubly stochastic)**.** *An $n \times n$ non-negative matrix is said to be doubly stochastic if all of its row and column sums are equal to 1.*

The matrix scaling problem is simple to describe: given an $n \times n$ non-negative matrix A, find a scaling of A which is doubly stochastic (if one exists). It turns out that an approximate version of the problem is more natural and has more structure, and as we will see in Section 3, this is not by accident. To define the approximate version, we will need another definition which is a quantification of how close a matrix is to being doubly stochastic.

Definition 2.3. *Given an $n \times n$ non-negative matrix A, define*

$$\text{ds}(A) = \sum_{i=1}^{n} (r_i - 1)^2 + \sum_{j=1}^{n} (c_j - 1)^2$$

Here r_i, c_j denote the i^{th} row and j^{th} column sums of A, respectively.

Now the goal in the ε-scaling problem is to find a scaling A' of A s.t. $\text{ds}(A') \leqslant \varepsilon$ (if one exists).

Definition 2.4 (Scalability)**.** *We will say that a non-negative matrix A is scalable if for all $\varepsilon > 0$, there exists a scaling A'_ε of A s.t. $\text{ds}(A'_\varepsilon) \leqslant \varepsilon$.*

Given this definition, several natural questions arise. When is a matrix scalable? If a matrix is scalable, can one efficiently find an ε-scaling? It turns out that answers to both questions are extremely pleasing! The answer to the first question is given by the following theorem (e.g. see [RS89]).

Theorem 2.5. *An $n \times n$ non-negative matrix A is scalable iff $\text{perm}(A) > 0$.*[1] *In other words, A is scalable iff the bipartite graph defined by the support of A has a perfect matching.*

What about the second question? If A is scalable, how to efficiently find and ε-scaling? Towards this Sinkhorn [Sin64] suggested an extremely natural algorithm which was analyzed in [LSW98].

[1] Here $\text{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} A_{i,\sigma}$ is the permanent of the matrix A.

[2] $r_1, \ldots, r_n$ denote the row sums of the current matrix i.e. BAC.

Input: $n \times n$ non-negative matrix A with rational entries of bit complexity at most b, and a distance parameter $\varepsilon > 0$.

Output: Either the algorithm correctly identifies that A is not scalable, or it outputs non-negative diagonal matrices B, C s.t. $\text{ds}(BAC) \leqslant \varepsilon$.

Algorithm:

1. Initialize $B = C = I$.
2. If A has an all-zero row or column, output not scalable and halt.
3. For T iterations, apply the following procedure:
   - If $\text{ds}(BAC) \leqslant \varepsilon$, output B, C and halt. If $\sum_{i=1}^{n}(r_i - 1)^2 > \varepsilon/2$,[2]normalize the rows i.e. $B \leftarrow \text{diag}(r_1^{-1}, \ldots, r_n^{-1})B$. Otherwise normalize the columns.
4. Output that A is not scalable.

Algorithm 1: Sinkhorn's algorithm

Theorem 2.6 ([LSW98]). *Algorithm 1 with $T = O(n(b+\log(n))/\varepsilon)$ iterations works correctly. That is, if the algorithm outputs A is not scalable, then A is not scalable. If A is scalable, then the algorithm will output an ε-scaling of A.*

It turns out that to test scalability, it suffices to take $\varepsilon = 1/(n+1)$. More formally,

Lemma 2.7 ([LSW98]). *Suppose A be an $n \times n$ non-negative matrix. If A is row or column normalized and $\text{ds}(A) \leqslant 1/(n+1)$, then A is scalable.*

Thus Algorithm 1 along with Theorem 2.5 and Theorem 2.6 gives an alternate (albeit slower) algorithm to test if a bipartite graph has a perfect matching.[3]

2.2 Operator scaling

The operator scaling problem was first studied by Gurvits [Gur04a]. We refer the reader to [Gur04a, GGOW16, AZGL+18] for various motivations, connections and applications. The objects of study here are tuples of $n \times n$ complex matrices $A = (A_1, \ldots, A_m)$. The name operator scaling comes from the fact that these tuples define a map from positive definite matrices to themselves, by $T_A(X) = \sum_{i=1}^{m} A_i X A_i^\dagger$.[4][5] But here we will restrict ourselves to the representation as tuple of matrices, for simplicity of exposition.

We start with a few definitions. First is the definition of scaling in this setting.

[3]Note the iterates in Algorithm 1 are row or column normalized and hence Lemma 2.7 applies.

[4]These maps are called completely positive maps/operators and are very natural from the point of view of quantum mechanics.

[5]$A_i^\dagger$ denotes the conjugate transpose of A_i.

Definition 2.8 (Scaling of tuples). *Given a tuple of $n \times n$ complex matrices, $A = (A_1, \ldots, A_m)$, we say that $A' = (A'_1, \ldots, A'_m)$ is a scaling of A if there exist invertible matrices B, C s.t. $A' = BAC$ i.e. $A'_i = BA_iC$ for all i.*

Next is the definition of doubly stochastic in this setting.

Definition 2.9 (Doubly stochastic tuples). *A tuple of $n \times n$ complex matrices, $A = (A_1, \ldots, A_m)$, is said to be doubly stochastic if*

$$\sum_{i=1}^{m} A_i A_i^\dagger = \sum_{i=1}^{m} A_i^\dagger A_i = I_n$$

As before, the operator scaling question is: given a tuple A, find a scaling which is doubly stochastic (if one exists). Again an approximate version is more natural. Towards that, we have the following definition quantifying how close a tuple is to being doubly stochastic.[6]

Definition 2.10. *Given a tuple of $n \times n$ complex matrices, $A = (A_1, \ldots, A_m)$, define*

$$\mathrm{ds}(A) = \left\| \sum_{i=1}^{m} A_i A_i^\dagger - I_n \right\|_F^2 + \left\| \sum_{i=1}^{m} A_i^\dagger A_i - I_n \right\|_F^2$$

Here $\|\cdot\|_F$ denotes the Frobenius norm.

The goal in the current version of ε-scaling problem is to find a scaling A' of A s.t. $\mathrm{ds}(A') \leqslant \varepsilon$ (if one exists).

Definition 2.11 (Scalability). *A tuple of $n \times n$ complex matrices, $A = (A_1, \ldots, A_m)$ is* scalable *if for all $\varepsilon > 0$, there exists a scaling A' of A s.t. $\mathrm{ds}(A') \leqslant \varepsilon$.*

We ask the same questions as before. When is a tuple scalable? If it is scalable, can we find an ε-scaling efficiently? There is a deep theory underlying the answer to the first question, and to unveil it we will need another definition.[7]

Definition 2.12 (Dimension non-decreasing tuples). *We say that a tuple of of $n \times n$ complex matrices, $A = (A_1, \ldots, A_m)$ is dimension non-decreasing if for all subspaces $V \subseteq \mathbb{C}^n$, $\dim\left(\sum_{i=1}^{m} A_i(V)\right) \geqslant \dim(V)$. Here $A_i(V)$ denotes the subspace $\{A_i v : v \in V\}$ and $V + W$ denotes the subspace $\mathrm{span}\{v + w : v \in V, w \in W\}$.*

The following theorem gives a very pleasing answer to the first question.

Theorem 2.13 ([Gur04a]). *A tuple of $n \times n$ complex matrices, $A = (A_1, \ldots, A_m)$ is scalable iff A is dimension non-decreasing.*

What about the second question? Gurvits [Gur04a] suggested a natural algorithm similar to that of Sinkhorn, although he could not analyze it in all cases. The full analysis, stated in the following theorem, was proved in [GGOW16].

---

[6] We apologize for the overload of notation. Some of it is deliberate to draw out the syntactic similarity between various scaling problems. As we will see later, there is a common thread that binds all these problems.

[7] Notice the similarity with the definition of dimension expanders (see [AFG14] and references therein).

[8] Here $BAC = (BA_1C, \ldots, BA_mC)$.

Input: A tuple of $n \times n$ matrices, $A = (A_1, \ldots, A_m)$ with entries in $\mathbb{Q}$ having bit complexity at most b and a distance parameter $\varepsilon > 0$.

Output: Either the algorithm correctly identifies that A is not scalable, or it outputs invertible matrices B, C s.t. $\mathrm{ds}(BAC) \leqslant \varepsilon$.[8]

Algorithm:

1. Initialize $B = C = I$.
2. If $\sum_{i=1}^m A_i A_i^\dagger$ or $\sum_{i=1}^m A_i^\dagger A_i$ is not invertible, output not scalable and halt.
3. Iterate for T iterations:
   - If $\mathrm{ds}(BAC) \leqslant \varepsilon$, output B, C and halt. If $\left\| \sum_{i=1}^m A_i A_i^\dagger - I_n \right\|_F^2 > \varepsilon/2$, left normalize i.e. $B \leftarrow \left(\sum_{i=1}^m A_i A_i^\dagger \right)^{-1/2} B$. Otherwise right normalize (which can be defined analogously).
4. Output that A is not scalable.

Algorithm 2: Gurvits' algorithm

Theorem 2.14 ([GGOW16]). *Algorithm 2 with $T = O(n(b + \log(n))/\varepsilon)$ iterations works correctly. That is if the algorithm outputs A is not scalable, then A is not scalable. If A is scalable, then the algorithm will output an ε-scaling of A.*

Similar to the matrix scaling setting, to test scalability, it suffices to take $\varepsilon = 1/(n+1)$. More formally,

Lemma 2.15 ([Gur04a]). *Suppose $A = (A_1, \ldots, A_m)$ is a tuple of $n \times n$ complex matrices. If $\sum_{i=1}^m A_i A_i^\dagger = I_n$ or $\sum_{i=1}^m A_i^\dagger A_i = I_n$, and $\mathrm{ds}(A) \leqslant 1/(n+1)$, then A is scalable.*

Hence Algorithm 2 along with Theorem 2.13 and Theorem 2.14 gives a polynomial time algorithm to test if a tuple is dimension non-decreasing.

Theorem 2.16 ([GGOW16]). *There is a polynomial time algorithm to test if a tuple $A = (A_1, \ldots, A_m)$ of $n \times n$ complex matrices is dimension non-decreasing.*

This was the first polynomial time algorithm for the operator scaling problem and as we will later see has applications in derandomization. Soon later, [IQS17a] (also see [IQS17b, DM15]) designed an algebraic algorithm for this problem which also works over finite fields. Their algorithm is an algebraic analogue of the augmenting paths algorithm for matching!

2.3 Tensor scaling

In this section, we will discuss a scaling problem which is a generalization of operator scaling and was studied in [BGO+18]. The objects of study here are tuples of tensors. Let

us denote the space of tensors $\mathbb{C}^{n_1} \otimes \cdots \otimes \mathbb{C}^{n_d}$ by $\mathrm{Ten}(n_1, \ldots, n_d)$. Then we will use the notation $A = (A_1, \ldots, A_m)$ to denote tuples of tensors where each $A_i \in \mathrm{Ten}(n_1, \ldots, n_d)$.

We start with the definition of scaling in this setting.[9]

Definition 2.17 (Tensor scaling of tuples)**.** *Given a tuple of tensors (in* $\mathrm{Ten}(n_1, \ldots, n_d)$*),* $A = (A_1, \ldots, A_m)$*, we say that* $A' = (A'_1, \ldots, A'_m)$ *is a scaling of* A *if there exist invertible matrices* $g_1, \ldots, g_d$ *s.t.* $A'_i = (g_1 \otimes \cdots \otimes g_d)A_i$ *for all* i*. We will use the notation* $A' = (g_1, \ldots, g_d) \cdot A$ *for this scaling action.*

Before going to the definition of stochastic tuples in this setting, we need to define a certain notion of marginals.

Definition 2.18 (Marginals)**.** *Given a tuple of tensors,* $A = (A_1, \ldots, A_m)$*, identify it with* $\mathrm{Ten}(m, n_1, \ldots, n_d)$*. Then we will denote the marginals of* A *by* $\rho_1^A, \ldots, \rho_d^A$*, where* $\rho_i^A \in M_{n_i, n_i}(\mathbb{C})$ *is a positive semidefinite matrix for all* i*. For each* i*, we can flatten* A *to obtain* $B_i \in M_{n_i, m \prod_{j \neq i} n_j}(\mathbb{C})$*. Then* $\rho_i^A = B_i B_i^\dagger$*. These are uniquely characterized by the following property:*

$$\mathrm{tr}\left[(I_m \otimes I_{n_1} \otimes \cdots M_i \otimes \cdots I_{n_d})AA^\dagger\right] = \mathrm{tr}\left[M_i \rho_i^A\right]$$

for all $M_i \in M_{n_i}(\mathbb{C})$ *and for all* $i \in [d]$*.*

Remark 2.19. *The above notion of marginals is very natural from the point of view of quantum mechanics. If one views* A *as representing a quantum state on* $d+1$ *systems indexed by* $0, 1, \ldots, d$*, then* $\rho_1^A, \ldots, \rho_d^A$ *are the marginals states on systems* $1, \ldots, d$ *respectively.*

Now we are ready to define the notion of stochasticity in this setting.

Definition 2.20 (d-stochastic tuples)**.** *A tuple of tensors (in* $\mathrm{Ten}(n_1, \ldots, n_d)$*)* $A = (A_1, \ldots, A_m)$ *is said to be* d*-stochastic if for each* i*,* $\rho_i^A = \frac{1}{n_i} I_{n_i}$ *i.e. the marginals are all scalar multiples of identity matrices.*

The normalization by $1/n_i$ is needed because $\mathrm{tr}\left[\rho_i^A\right] = \|A\|_2^2$ for all i. We will also need the following measure which quantifies how close a tuple is to being d-stochastic.

Definition 2.21. *Given a tuple of tensors,* $A = (A_1, \ldots, A_m)$*, define*

$$\mathrm{ds}(A) = \sum_{i=1}^{d} \left\| \rho_i^A - \frac{1}{n_i} I_{n_i} \right\|_F^2$$

Note that the definition above differs from Definitions 2.3 and 2.10 slightly in terms of a normalization factor. As before the ε-scaling problem is to find a scaling A' of A s.t. $\mathrm{ds}(A') \leqslant \varepsilon$ (if one exists). Scalability is also defined similarly as before.

Definition 2.22 (Scalability)**.** *We will say that a tuple of tensors,* $A = (A_1, \ldots, A_m)$*, is scalable if for all* $\varepsilon > 0$*, there exists a scaling* A' *of* A *s.t.* $\mathrm{ds}(A') \leqslant \varepsilon$*.*

The same questions arise. When is a tuple scalable? If it is scalable, can one find an ε-scaling efficiently? The answer to the first question is given by remarkable and deep theorems of Hilbert and Mumford, and Kempf and Ness. To properly state this answer we need some more definitions.

[9]The scaling here looks very different from matrix scaling. One can also define a generalization of matrix scaling to tensors but we will not focus on that version in this survey (see [FL89]).

Definition 2.23 (Deficiency). *We call a subset $S \subseteq [n_1] \times \cdots \times \cdots [n_d]$ deficient if there exist real numbers $(a_{i,j})_{i\in[d],j\in[n_i]}$ s.t. $\sum_{i=1}^{d} a_{i,j_i} > 0$ for all $(j_1, \ldots, j_d) \in S$.*

We encourage the reader to work out an alternate characterization of deficiency in the case $d = 2$ and $n_1 = n_2$. Hint: it is related to perfect matchings in bipartite graphs.

We will also use the following notation.

$$\text{supp}(A) = \{(j_1, \ldots, j_d) \in [n_1] \times \cdots \times [n_d] : \exists i \in [m] \text{ s.t. } A_i(j_1, \ldots, j_d) \neq 0\}$$

Theorem 2.24 (Hilbert-Mumford + Kempf-Ness [Hil93, Mum65, KN79], see [BGO+18]). *A tuple of tensors $A = (A_1, \ldots, A_m)$ is scalable iff for every tuple of invertible matrices $(g_1, \ldots, g_d)$, $\text{supp}((g_1, \ldots, g_d) \cdot A)$ is not deficient.*

We leave it as an exercise to verify that the above theorem is the same as Theorem 2.13 in the case $d = 2$ and $n_1 = n_2$.

How to find an efficient scaling if one exists? It turns out that one can extend the alternating minimization kind of algorithms from the matrix and operator scaling settings to the tensor scaling setting as well.

Input: A tuple of tensors (in $\text{Ten}(n_1, \ldots, n_d)$), $A = (A_1, \ldots, A_m)$ with entries in $\mathbb{Q}$ having bit complexity at most b and a distance parameter $\varepsilon > 0$.

Output: Either the algorithm correctly identifies that A is not scalable, or it outputs invertible matrices $(g_1, \ldots, g_d)$ s.t. $\text{ds}((g_1, \ldots, g_d) \cdot A) \leqslant \varepsilon$.[10]

Algorithm:

1. Initialize $g_i = I_{n_i}$.
2. If for some i, ρ_i^A is not invertible, output not scalable and halt.
3. Iterate for T iterations:
   - If $\text{ds}((g_1, \ldots, g_d) \cdot A) \leqslant \varepsilon$, output $(g_1, \ldots, g_d)$ and halt. If for some i, $\left\|\rho_i^A - I_{n_i}\right\|_F^2 > \varepsilon/d$, normalize the i^{th} coordinate i.e. $g_i \leftarrow \left(n_i \rho_i^A\right)^{-1/2} g_i$.
4. Output that A is not scalable.

Algorithm 3: Tensor scaling algorithm

Algorithm 3 was proposed in [VDDM03] without analysis. The following theorem regarding the analysis of the algorithm was proved in [BGO+18].

Theorem 2.25 ([BGO+18]). *Algorithm 2 with $T = O(d(b + \log(mn_1 \cdots n_d))/\ell\varepsilon)$ iterations works correctly ($\ell = \min_i n_i$). That is if the algorithm outputs A is not scalable, then A is not scalable. If A is scalable, then the algorithm will output an ε-scaling of A.*

Unfortunately, unlike the matrix and operator scaling case, to test scalability, it is not sufficient to take ε which is polynomially small (see [BGO+18] for a discussion). Hence we still do not have a polynomial time algorithm for testing scalability of tensors.

[10] Here $(g_1, \ldots, g_d) \cdot A = ((g_1 \otimes \cdots \otimes g_d)A_1, \ldots, (g_1 \otimes \cdots \otimes g_d)A_m)$.

3 Source of scaling

Given the syntactic similarities between Sections 2.1, 2.2 and 2.3, it is natural to wonder if there is a general setting which captures all these scaling problems. In other words, where does scaling come from? It turns out that scaling arises in an algebraic setting and understanding the algebraic setting is crucial to a unified analysis of Algorithms 1, 2 and 3.

In Section 3.1, we introduce basic concepts in invariant theory, which provides crucial tools for the analysis of scaling algorithms. In Section 3.2 we introduce basic concepts of geometric invariant theory, which elucidates the connection between invariant theory and scaling problems.

3.1 Invariant theory: source of scaling

Invariant theory studies the linear actions of groups on vector spaces. We refer the reader to the excellent books [DK15, Stu08] for an extensive introduction to the area. We will only cover a few basics that we need for our purpose here. Invariant theory deals with linear actions of groups on vector spaces. For our purpose vector spaces will be over complex numbers ($\mathbb{C}$) and the groups we will deal with will be extremely simple - special linear group, denoted by $\mathrm{SL}(n)$ ($n \times n$ matrices over $\mathbb{C}$ with determinant 1), direct products of special linear group as well as the diagonal subgroup of the special linear group, denoted by $\mathrm{ST}(n)$ (diagonal $n \times n$ matrices over $\mathbb{C}$ with determinant 1) and direct products. However the theory is quite general and generalizes to large class of groups.

Suppose we have a group G which acts linearly on a vector space V.[11] Fundamental objects of study in invariant theory are the invariant polynomials which are just polynomial functions on V left invariant by the action of the group G. Invariant polynomials form a ring and this ring is usually denoted by $\mathbb{C}[V]^G$. More formally,

$$\mathbb{C}[V]^G = \{p \in \mathbb{C}[V] : p(g \cdot v) = p(v) \; \forall \; g \in G, v \in V\}$$

Let us consider a simple example. The group $G = \mathrm{SL}(n) \times \mathrm{SL}(n)$ acts on the vector space $V = \mathrm{Mat}_n(\mathbb{C})$ [12] by left-right multiplication as follows: $(A, B) \cdot X = AXB^T$. $\mathrm{Det}(X)$ is an invariant polynomial for this action and it turns out it is the only one (prove it!). That is any invariant polynomial is just of the form $q(\mathrm{Det}(X))$, for a univariate polynomial q, or in other words, $\mathrm{Det}(X)$ generates the invariant ring. As an aside (this will not be so important for us), Hilbert [Hil90, Hil93] proved that the invariant ring is always finitely generated! [13] These papers proved several theorems which are the building blocks of modern algebra, like Nullstellansatz and finite basis theorem, as “lemmas" enroute to proving the finite generation of invariant rings!

[11] That is the group action satisfies the following axioms: $g \cdot (c_1 v_1 + c_2 v_2) = c_1 g \cdot v_1 + c_2 g \cdot v_2$ for all $c_1, c_2 \in \mathbb{C}$ and $v_1, v_2 \in V$, in addition to the properties of being a group action i.e. $g_1 \cdot (g_2 \cdot v) = (g_1 g_2) \cdot v$, $e \cdot v = v$ for all $g_1, g_2 \in G$, $v \in V$ and for e being the identity element of the group. Usually one also requires that the action is algebraic.

[12] $\mathrm{Mat}_n(\mathbb{C})$ denotes the space of $n \times n$ complex matrices

[13] He proved it for the actions of general linear groups but his proof readily generalizes to a more general class of groups called reductive groups.

Some other fundamental objects of study in invariant theory are orbits and orbit-closures. The orbit of a vector $v \in V$, $\mathcal{O}_G(v)$ is simply the set of all vector elements that v can be transformed to by the group action. That is,

$$\mathcal{O}_G(v) = \{g \cdot v : g \in G\}$$

An orbit-closure, $\overline{\mathcal{O}}_G(v)$ of a vector v is obtained by simply including all the limit points of sequences of points in an orbit. That is,

$$\overline{\mathcal{O}}_G(v) = \{w \in V : \exists g_1, \ldots, g_k, \ldots, \text{ s.t. } \lim_{k\to\infty} g_k \cdot v = w\}$$

Many important problems in theoretical computer science are really questions about orbit-closures. To list a few,

1. The graph isomorphism problem is about checking if the orbit closures[14] of two graphs (under the group action of permuting the vertices) are the same or not.

2. The $\mathcal{VP}$ vs $\mathcal{VNP}$ question (or more precisely a variant of it) can be phrased as testing if the (padded) permanent polynomial lies in the orbit-closure of the determinant (w.r.t. the action on the polynomials induced by the action of general linear group on the variables). This is the starting point of geometric complexity theory (GCT) [MS02, Bür12, Lan15].

3. The question of tensor rank lower bounds (more precisely border rank) can be phrased as asking if a padded version of the given tensor lies in the orbit-closure of the diagonal unit tensor (w.r.t. the natural action of products of general linear groups on the tensors). This approach also falls under the purview of geometric complexity theory [BI11, BI13].

It turns out that a very simple concept in invariant theory captures the mysteries about the scaling problems in Sections 2.1, 2.2 and 2.3. This is the so called *null cone* of a group action (on a vector space). The null cone has dual definitions in terms of the invariant polynomials as well as orbit-closures (in a very general setting, and in particular for the group actions we care about in this survey). This duality is quite important for the analysis of the scaling algorithms.

Definition 3.1 (Null cone)**.** *The null cone for a group G acting on a vector space V, denoted by $\mathcal{N}_G(V)$, is the zero set of all homogeneous invariant polynomials. That is,*

$$\mathcal{N}_G(V) = \{v \in V : p(v) = 0 \; \forall \textit{ homogeneous } p \in \mathbb{C}[V]^G\}$$

It is a cone since $v \in \mathcal{N}_G(V)$ implies that $cv \in \mathcal{N}_G(V)$ for all $c \in \mathbb{C}$. A theorem due to Hilbert [Hil93] and Mumford [Mum65] [15] says that for a large class of group actions (which includes the group actions we will study), $v \in \mathcal{N}_G(V)$ iff $0 \in \overline{\mathcal{O}}_G(v)$ (try to figure out the easy direction). This is a consequence of Hilbert's Nullstellensatz along with the fact that orbit-closures for certain group actions are algebraic varieties (or in other words Euclidean and Zariski closures match). If we look at the left-right multiplication

[14] Note that for the action of a finite group, the orbit of a point is the same as its orbit closure.

[15] Not to be confused with Hilbert-Mumford criterion which we will come across later.

example discussed above, the null cone is just the space of singular matrices since determinant generates the invariant ring. We leave it as an exercise to verify that the 0 matrix lies in the orbit-closure of any singular matrix (under the left-right multiplication action of $\mathrm{SL}(n) \times \mathrm{SL}(n)$).

We will now describe the connection between null cone and scaling problems. For this we will need to move on to the area of geometric invariant theory, which provides geometric and analytic tools to study problems in invariant theory, and also provides with an intriguing non-commutative extension of Farkas' lemma (or linear programming duality). As a teaser of things to come, the objects in Sections 2.1, 2.2 and 2.3 are scalable iff they are not in the null cone of certain group actions!

3.2 Geometric invariant theory: non-commutative duality

In this section, we will give a brief overview of the geometric invariant theoretic approach to studying the null cone problem. This will also fit in nicely with the computational aspects of the null cone. Section 3.2.1 describes the Hilbert-Mumford criterion which is really answering the question: how does one prove if some vector is in the null cone. Section 3.2.2 describes Kempf-Ness which answers the question: how does one prove if some vector is not in the null cone. Section 3.2.3 studies the Hilbert-Mumford and the Kempf-Ness criterion for certain commutative group actions and explains why these generalize Farkas' lemma. Section 3.3 explains the connection between geometric invariant theory and scaling problems.

3.2.1 Hilbert-Mumford criterion

Fix the action of a group G on a vector space V. How does one prove to someone that a vector v is in the null cone? We know that v is in the null cone iff $0 \in \overline{\mathcal{O}}_G(v)$ i.e. there is a sequence of group elements $g_1, \ldots, g_k, \ldots$ s.t. $\lim_{k\to\infty} g_k \cdot v = 0$. So this sequence of group elements is a witness to v being in the null cone. Is their a more succinct witness? After all, how do we even describe an infinite sequence of group elements? The Hilbert-Mumford criterion says that there does exist a more succinct witness (again we won't go into the technical conditions the group G needs to satisfy but just say that they will be satisfied for the groups we will consider).

Theorem 3.2 (Hilbert-Mumford criterion [Hil93, Mum65])**.** *$v \in \mathcal{N}_G(V)$ iff there is a one-parameter subgroup λ of G s.t. $\lim_{t\to 0} \lambda(t) \cdot v = 0$.*

What this means is that instead of looking at all sequences of group elements, one only needs to restrict our attention to those sequences of group elements which can be succinctly described by one-parameter subgroups. What are one-parameter subgroups? These are just algebraic group homomorphisms (i.e. an algebraic map which is also a group homomorphism) $\lambda : \mathbb{C}^* \to G$. Let us look at several examples (we encourage the reader to prove these statements).

1. For the group $G = \mathbb{C}^*$ (the multiplicative group of non-zero complex numbers), all one parameter subgroups are of the form $\lambda(t) = t^a$ for some $a \in \mathbb{Z}$.

2. For the group $G = (\mathbb{C}^*)^n$ (direct product of n copies of $\mathbb{C}^*$), all one parameter subgroups are of the form $\lambda(t) = (t^{a_1}, \ldots, t^{a_n})$ for some $(a_1, \ldots, a_n) \in \mathbb{Z}^n$.

3. For the group $G = \mathrm{ST}(n)$ (diagonal $n \times n$ matrices with determinant 1), all one parameter subgroups are of the form $\lambda(t) = (t^{a_1}, \ldots, t^{a_n})$ for some $(a_1, \ldots, a_n) \in \mathbb{Z}^n$ satisfying $\sum_{i=1}^n a_i = 0$.

4. For the group $G = \mathrm{ST}(n) \times \mathrm{ST}(n)$, all one parameter subgroups are of the form

$$\lambda(t) = \left((t^{a_1}, \ldots, t^{a_n}), \left(t^{b_1}, \ldots, t^{b_n}\right)\right)$$

   for some $(a_1, \ldots, a_n), (b_1, \ldots, b_n) \in \mathbb{Z}^n$ satisfying $\sum_{i=1}^n a_i = \sum_{i=1}^n b_i = 0$.

5. For the group $G = \mathrm{GL}(n)$ ($n \times n$ invertible matrices), all one parameter subgroups are of the form $\lambda(t) = S \,\mathrm{diag}\,(t^{a_1}, \ldots, t^{a_n})\, S^{-1}$ for some $S \in \mathrm{GL}(n)$ and some $(a_1, \ldots, a_n) \in \mathbb{Z}^n$. Here $\mathrm{diag}\,(t_1, \ldots, t_n)$ represents a diagonal matrix with $(t_1, \ldots, t_n)$ on the diagonal.

6. For the group $G = \mathrm{SL}(n)$ ($n \times n$ invertible matrices with determinant 1), all one parameter subgroups are of the form $\lambda(t) = S \,\mathrm{diag}\,(t^{a_1}, \ldots, t^{a_n})\, S^{-1}$ for some $S \in \mathrm{SL}(n)$ and some $(a_1, \ldots, a_n) \in \mathbb{Z}^n$ satisfying $\sum_{i=1}^n a_i = 0$.

7. For the group $G = \mathrm{SL}(n_1) \times \cdots \mathrm{SL}(n_d)$, all one parameter subgroups are of the form

$$\lambda(t) = \left(S_1 \,\mathrm{diag}\,(t^{a_{1,1}}, \ldots, t^{a_{1,n_1}})\, S_1^{-1}, \ldots, S_d \,\mathrm{diag}\,(t^{a_{d,1}}, \ldots, t^{a_{d,n_d}})\, S_d^{-1}\right)$$

   for some $S_i \in \mathrm{SL}(n_i)$ and some integer $a_{i,j}$'s satisfying $\sum_{j=1}^{n_i} a_{i,j} = 0$ for all $i \in [d]$.

Let us return to the example of the left-right multiplication action of $G = \mathrm{SL}(n) \times \mathrm{SL}(n)$ on $V = \mathrm{Mat}_n(\mathbb{C})$. Recall that (A, B) sends M to AMB^T and M is in the null cone iff it is singular. If M is singular, what is a one-parameter subgroup driving it to the zero matrix? Since M is singular, there exists an invertible S (which can be taken to have determinant 1) s.t. $S^{-1}M$ has the last row all zeroes. Then the one-parameter subgroup

$$\lambda(t) = \left(S \,\mathrm{diag}\left(t, t, \ldots, t, t^{-(n-1)}\right) S^{-1}, I_n\right)$$

sends M to the zero matrix. Later we will see more examples corresponding to each of the scaling problems.

Having understood how to prove if a given vector is in the null cone, we move on to study how to prove that a given vector is *not* in the null cone.

3.2.2 Kempf-Ness theorem

Fix the action of a group G on a vector space V. How does one prove that a vector v is not in the null cone? We know that a vector $v \notin \mathcal{N}_G(V)$ iff there is a homogeneous invariant polynomial p s.t. $p(v) \neq 0$. Such a p can serve as a witness that $v \notin \mathcal{N}_G(V)$. However, these polynomials typically have exponentially large degree (see [Der01]) and may not have any efficient description. An alternative witness is given by the Kempf-Ness theorem [KN79].

To state the Kempf-Ness theorem, we need to (informally) define something called a moment map, which relies on the following function,

$$f_v(g) = \|g \cdot v\|_2^2$$

This function defines the following optimization problem,

$$\mathcal{N}(v) = \inf_{g \in G} f_v(g) \tag{1}$$

Note that $v \notin \mathcal{N}_G(V)$ iff $\mathcal{N}(v) > 0$. Now the moment map at v, denoted by $\mu_G(v)$ is simply the gradient of the function f_v "along the group action" at $g = e$ (the identity element of the group G).[16] We will not go into the specifics of the space in which $\mu_G(v)$ lives but instead do the moment map calculation for several examples. First let us state the Kempf-Ness theorem.

Theorem 3.3 (Kempf-Ness [KN79]). *$v \notin \mathcal{N}_G(V)$ iff there is a non-zero $w \in \overline{\mathcal{O}}_G(v)$ s.t. $\mu_G(w) = 0$.*

If $v \notin \mathcal{N}_G(V)$, then there exists a non-zero $w \in \overline{\mathcal{O}}_G(v)$ which is of minimal norm and hence $\mu_G(w) = 0$. So this is the easy direction. The amazing part about the Kempf-Ness theorem is that any local minima becomes a global minima i.e. if $\mu_G(w) = 0$ for some non-zero $w \in \overline{\mathcal{O}}_G(v)$, then $v \notin \mathcal{N}_G(V)$, even though $\mu_G(w) = 0$ only guarantees that one cannot decrease the norm of w by actions of group elements close to identity (that is, "local" group actions). This smells of some kind of convexity and indeed, the function $f_v(g)$ is geodesically convex (i.e. convex w.r.t. some appropriate metric on the group). We will not delve more into geodesic convexity or moment maps in this survey but refer the interested reader to [NM84, Woo11, HH12, GRS13].

Let us return to the example of the left-right multiplication action of $G = \mathrm{SL}(n) \times \mathrm{SL}(n)$ on $V = \mathrm{Mat}_n(\mathbb{C})$. Recall that (A, B) sends M to AMB^T and M is in the null cone iff it is singular. What is the moment map in this case? $\mu_G(M) = (P_1, P_2)$, where P_1, P_2 are $n \times n$ traceless matrices s.t.

$$\begin{aligned}\mathrm{tr}\left[P_1^T Q_1\right] + \mathrm{tr}\left[P_2^T Q_2\right] &= \frac{d}{ds}\left\|\exp\left(sQ_1\right) M \exp\left(sQ_2^T\right)\right\|_F^2 \Big|_{s=0} \\ &= 2\,\mathrm{tr}[MM^\dagger Q_1] + 2\,\mathrm{tr}\left[\left(M^\dagger M\right)^T Q_2\right]\end{aligned}$$

for all Hermitian traceless matrices Q_1, Q_2. [17] Here $M^\dagger$ denotes the conjugate transpose of the matrix M. Thus

$$P_1 = \left(MM^\dagger\right)^T - \frac{\|M\|_F^2}{n} I_n \text{ and } P_2 = M^\dagger M - \frac{\|M\|_F^2}{n} I_n$$

Hence $\mu_G(M) = 0$ is the same as saying M is a scalar multiple of a unitary matrix. It is not hard to see that any non-singular M can be brought to such a form by the left-right multiplication action.

In the next section, we will see what Hilbert-Mumford and Kempf-Ness theorem look like for actions of $\mathrm{T}(n) = (\mathbb{C}^*)^n$. Readers wanting to get to the setting of scaling problems could skip the next section.

[16] There are minor differences between this definition and how moment map is usually defined.

[17] It suffices to focus on Hermitian matrices.

3.2.3 Commutative group actions: Farkas' lemma

In this section, we play around with the Hilbert-Mumford criterion and Kempf-Ness theorem and see what it gives for actions of the group $G = \mathrm{T}(n) = (\mathbb{C}^*)^n$.

Fix vectors $\omega^{(1)}, \ldots, \omega^{(m)} \in \mathbb{Z}^n$. Then G acts on $V = \mathbb{C}^m$ as follows: $(t_1, \ldots, t_n)$ sends the i^{th} basis vector e_j to $\prod_{i=1}^n t_i^{\omega_i^{(j)}} e_j$. That is e_j is an eigenvector of the action of $(t_1, \ldots, t_n)$ with eigenvalue $\prod_{i=1}^n t_i^{\omega_i^{(j)}}$. We urge the reader to prove that all actions of G look essentially like this.

What is the null cone for this action? Let us apply Hilbert-Mumford criterion. Recall that all the one-parameter subgroups of G look like $\lambda(t) = (t^{a_1}, \ldots, t^{a_n})$ for some $(a_1, \ldots, a_n) \in \mathbb{Z}^n$. Now fix $v \in V$, where $v = \sum_{j=1}^m v_j e_j$, with $v_j \in \mathbb{C}$, and denote by $\mathrm{supp}(v)$, the support of v i.e.

$$\mathrm{supp}(v) = \{j \in [m] : v_j \neq 0\}$$

Then the Hilbert-Mumford criterion (Theorem 3.2) tells us that $v \in \mathcal{N}_G(V)$ iff there is a one-parameter subgroup λ that drives v to zero. That is, there exists $a = (a_1, \ldots, a_n) \in \mathbb{Z}^n$ s.t.

$$\lim_{t\to 0} \prod_{i=1}^n t^{a_i \omega_i^{(j)}} = \lim_{t\to 0} t^{\langle a, \omega^{(j)} \rangle} = 0$$

for all $j \in \mathrm{supp}(v)$. Equivalently, we have:

Proposition 3.4. *$v \in \mathcal{N}_G(V)$ iff there exists $a \in \mathbb{Z}^n$ s.t. $\langle a, \omega^{(j)} \rangle > 0$ for all $j \in \mathrm{supp}(v)$.*

Now let us see what the Kempf-Ness theorem says in this setting. By computing the moment map $\mu_G(v) \in \mathbb{R}^n$ we see that it satisfies the following,

$$\begin{aligned} \langle \mu_G(v), b \rangle &= \frac{d}{ds} \left\| (\exp(sb_1), \ldots, \exp(sb_n)) \cdot v \right\|_2^2 \Big|_{s=0} \\ &= 2 \sum_{j=1}^m |v_j|^2 \langle \omega^{(j)}, b \rangle \end{aligned}$$

for all $b \in \mathbb{R}^n$. Hence $\mu_G(v) = 2\sum_{j=1}^m |v_j|^2 \omega^{(j)} = 2\sum_{j \in \mathrm{supp}(v)} |v_j|^2 \omega^{(j)}$. Now the Kempf-Ness theorem (Theorem 3.3) says that $v \notin \mathcal{N}_G(V)$ iff there exists non-zero $w \in \overline{\mathcal{O}}_G(v)$ s.t. $\mu_G(w) = 0$. Note that if there exists non-zero $w \in \overline{\mathcal{O}}_G(v)$ s.t. $\mu_G(w) = 0$, then $0 \in \mathrm{conv}\left(\left(\omega^{(j)}\right)_{j \in \mathrm{supp}(v)} \right)$. So this matches with the conclusions of the Farkas' lemma which says that there exists $a \in \mathbb{Z}^n$ s.t. $\langle a, \omega^{(j)} \rangle > 0$ for all $j \in \mathrm{supp}(v)$ iff $0 \notin \mathrm{conv}\left(\left(\omega^{(j)}\right)_{j \in \mathrm{supp}(v)} \right)$. The first part of the Farkas' lemma matches the case $v \in \mathcal{N}_G(V)$ via the Hilbert-Mumford criterion and the second part matches the case $v \notin \mathcal{N}_G(V)$ via the Kempf-Ness theorem!

3.3 Hilbert-Mumford, Kempf-Ness and scaling

In this section, we delve into the connection between geometric invariant theory and various scaling problems. Sections 3.3.1, 3.3.2 and 3.3.3 consider the consequences of Hilbert-Mumford and Kemp-Ness theorems for the matrix, operator and tensor scaling problems, respectively.

3.3.1 Matrix scaling

We elucidate here the connection between geometric invariant theory and matrix scaling. For the connection to invariant theory, we need a group action on a vector space. Given that the objects of study are $n \times n$ non-negative real matrices, it is natural to guess the vector space would be $V = \mathrm{Mat}_n(\mathbb{C})$ (given that we only discussed invariant theory with the base field being $\mathbb{C}$). But what is the group action? The group action is also almost given away by the definition of scaling. The first guess might be that the group is $G = \mathrm{T}(n) \times \mathrm{T}(n)$ and it acts on $V = \mathrm{Mat}_n(\mathbb{C})$ as follows,

$$((t_1, \ldots, t_n), (s_1, \ldots, s_n)) \cdot M = \mathrm{diag}(t_1, \ldots, t_n)\, M\, \mathrm{diag}(s_1, \ldots, s_n)$$

But it turns out that the null cone for this action is the whole of V (verify this). But the above guess comes pretty close and the right thing is obtained by looking at an appropriate normalization. It turns out that the group G would be $\mathrm{ST}(n) \times \mathrm{ST}(n)$ and it acts on V by the same action as above (it won't be immediately clear why imposing a determinant 1 constraint on the group elements is the right thing to do).

Now let us see what the Hilbert-Mumford criterion (Theorem 3.2) and Kempf-Ness theorem (Theorem 3.3) say about this group action.

Recall from Section 3.2.1 that all the one-parameter subgroups of G look like

$$\lambda(t) = \left(\left(t^{a_1}, \ldots, t^{a_1}\right), \left(t^{b_1}, \ldots, t^{b_n}\right)\right)$$

for some integers $a_1, \ldots, a_n, b_1, \ldots, b_n$ satisfying $\sum_{i=1}^n a_i = \sum_{i=1}^n b_i = 0$. Let us denote by $\mathrm{supp}(M)$, the support of M i.e.

$$\mathrm{supp}(M) = \{(i, j) \in [n] \times [n] : M_{i,j} \neq 0\}$$

Then the Hilbert-Mumford criterion says that M is in the null cone iff there exists a one-parameter subgroup λ as above s.t.

$$\lim_{t \to 0} \lambda(t) \cdot M = 0$$

Equivalently,

Corollary 3.5. *$M \in \mathcal{N}_G(V)$ iff there exist integers $a_1, \ldots, a_n, b_1, \ldots, b_n$ satisfying $\sum_{i=1}^n a_i = \sum_{i=1}^n b_i = 0$ s.t. $a_i + b_j > 0$ for all $(i, j) \in \mathrm{supp}(M)$.*

We encourage the reader to prove that the above proposition implies that $M \in \mathcal{N}_G(V)$ iff the bipartite graph defined by $\mathrm{supp}(M)$ has no perfect matching.

Now let us apply the Kempf-Ness theorem. First let us calculate the moment map. $\mu_G(M) = (p, q)$, where $p, q \in \mathbb{R}^n$ and $\sum_{i=1}^n p_i = \sum_{j=1}^n q_j = 0$ and it satisfies the following,

$$\begin{aligned}
\langle p, d \rangle + \langle q, e \rangle &= \frac{d}{ds} \left\| \left(\left(\exp(sd_1), \ldots, \exp(sd_n)\right), \left(\exp(se_1), \ldots, \exp(se_n)\right) \right) \cdot M \right\|_F^2 \Bigg|_{s=0} \\
&= 2 \sum_{i,j} |M|_{i,j}^2 (d_i + e_j) \\
&= 2\langle r_M, d \rangle + 2\langle c_M, e \rangle
\end{aligned}$$

for all $d, e \in \mathbb{R}^n$ satisfying $\sum_{i=1}^n d_i = \sum_{j=1}^n e_j = 0$. Here r_M and c_M are the vectors of row and column sums of the matrix $(|M_{i,j}|^2)_{i\in[n],j\in[n]}$, respectively. Thus $p = r_M - \text{avg}_M \mathbf{1}$ and $q = c_M - \text{avg}_M \mathbf{1}$, where

$$\text{avg}_M = \sum_{i=1}^n r_M(i)/n = \sum_{j=1}^n c_M(j)/n$$

and $\mathbf{1}$ is the all 1's vector. Now the Kempf-Ness theorem says that $M \notin \mathcal{N}_G(V)$ iff there exists a non-zero $N \in \overline{\mathcal{O}}_G(M)$ s.t. $\mu_G(N) = 0$. Equivalently,

Corollary 3.6. *$M \notin \mathcal{N}_G(V)$ iff the non-negative real matrix A_M, given by $A_M(i,j) = |M_{i,j}|^2$, is scalable.*

Corollaries 3.5 and 3.6 together yield a proof of Theorem 2.5.

3.3.2 Operator scaling

For the operator scaling problem, the vector space is clear, $V = \text{Mat}_n(\mathbb{C})^m$, i.e. m copies of $\text{Mat}_n(\mathbb{C})$. The group action is also clear (except for the normalization to determinant 1). $G = \text{SL}(n) \times \text{SL}(n)$ and it acts on V as follows,

$$(B, C) \cdot (A_1, \ldots, A_m) = (BA_1C^T, \ldots, BA_mC^T)$$

This action is sometimes called the left-right action. We leave the details of the Hilbert-Mumford criterion and Kempf-Ness theorem to the reader and only say that they yield the following corollaries which together imply Theorem 2.13.

Corollary 3.7 (Hilbert-Mumford for left-right action)**.** *$A = (A_1, \ldots, A_m) \in \mathcal{N}_G(V)$ iff A is dimension non-decreasing (Definition 2.12).*

Corollary 3.8 (Kempf-Ness for left-right action)**.** *$A = (A_1, \ldots, A_m) \notin \mathcal{N}_G(V)$ iff A is scalable.*

3.3.3 Tensor scaling

For the tensor scaling problem, the vector space is $V = \text{Ten}(n_1, \ldots, n_d)^m$. The group is $G = \text{SL}(n_1) \times \cdots \times \text{SL}(n_d)$ which acts on V as follows,

$$(g_1, \ldots, g_d) \cdot (A_1, \ldots, A_m) = ((g_1 \otimes \cdots \otimes g_d)\, A_1, \ldots, (g_1 \otimes \cdots \otimes g_d)\, A_m)$$

Again we will leave the details of the Hilbert-Mumford criterion and Kempf-Ness theorem to the reader and only say that they yield the following corollaries, which together imply Theorem 2.24.

Corollary 3.9 (Hilbert-Mumford for tensor action)**.** *$A = (A_1, \ldots, A_m) \in \mathcal{N}_G(V)$ iff there is a tuple of invertible matrices (of appropriate sizes) $(g_1, \ldots, g_d)$ s.t. supp $((g_1, \ldots, g_d) \cdot A)$ is deficient (Definition 2.23).*

Corollary 3.10 (Kempf-Ness for tensor action)**.** *$A = (A_1, \ldots, A_m) \notin \mathcal{N}_G(V)$ iff A is scalable.*

4 Analysis of scaling algorithms

In this section, we provide a unified analysis of the scaling algorithms described in Sections 2.1, 2.2 and 2.3. We will first design a common template and analysis for Algorithms 1, 2 and 3 and then look at each case separately to fill in the details that need to be done differently. Most of the analysis will be common and the only difference will be the choice of a potential function (although the source of all the potential functions will be invariant theory). Algorithm 4 contains a common template for all the three scaling algorithms.

Let us now turn to performing an analysis of Algorithm 4. We need a potential function and the source of potential functions will be invariant theory. Recall from Section 3.3 that A is a scalable iff $A \notin \mathcal{N}_G(V)$ (Corollaries 3.6, 3.8 and 3.10).[19] Here G is a subgroup of $\widetilde{G}$, and $G = \mathrm{ST}(n) \times \mathrm{ST}(n)$, $\mathrm{SL}(n) \times \mathrm{SL}(n)$ or $\mathrm{SL}(n_1) \times \cdots \times \mathrm{SL}(n_d)$ for matrix, operator or tensor scaling, respectively. We also know from Definition 3.1 that $A \notin \mathcal{N}_G(V)$ iff there exists an $\ell \in \mathbb{N}$ and a homogeneous polynomial of degree ℓ that is invariant under the action of G s.t. $P(A) \neq 0$. Suppose there exists such a P that has integer coefficients and satisfies

$$|P(A)| \leqslant U^{\ell} \|A\|^{\ell} \tag{2}$$

Then we we will prove the following theorem regarding the analysis of Algorithm 4.

Theorem 4.1 (Unified analysis of scaling algorithms)**.** *If A is scalable, then running Algorithm 4 for $T = O((\log(U) + b)/\varepsilon'')$ iterations suffices to output a scaling A' s.t. $\widetilde{\mathrm{ds}}(A') \leqslant \varepsilon$. Here $\varepsilon'' = n\varepsilon$ for matrix and operator scaling, and $\varepsilon'' = (\min_i n_i)\,\varepsilon/d$ for tensor scaling.*

The analysis will be a three step analysis that is common to a lot of the scaling papers [GY98, LSW98, Gur04a, GGOW16, BGO+18, Fra18, BFG+18]. We will need the following lemma for [LSW98] which is essentially a robust version of the AM-GM inequality.

Lemma 4.2. *Let $x_1, \ldots, x_n$ be positive real numbers s.t. $\sum_{i=1}^n x_i = n$ and $\sum_{i=1}^n (x_i - 1)^2 = \delta \leqslant 1$. Then,*

$$\prod_{i=1}^{n} x_i \leqslant \exp(-\delta/6)$$

Lemma 4.2 implies that for $\widehat{h}$ as in Algorithm 4,

$$\mathrm{Det}\left(\widehat{h}\right)^{1/n'} \geqslant \exp(-\varepsilon'/12n') \tag{3}$$

Now the three steps of the analysis are as follows. The potential function is $\Phi(A) = |P(A)|^{1/\ell}$, and its explicit description will be given in the next subsections.

1. **Lower bound**: Since $P(A) \neq 0$, P is a homogeneous polynomial of degree ℓ with integer coefficients and A has rational entries with bit complexity at most b, it follows that $|P(A)| \geqslant 2^{-b\ell}$ and hence $\Phi(A) \geqslant 2^{-b}$.

[18] This is to reconcile the minor differences in the definition of ds for various measures.

[19] Corollary 3.6 says something slightly different but the variant we state here is true as well.

Input: Object A which has norm $\|A\| = 1$, with all entries having bit complexity at most b and distance parameter $\varepsilon > 0$. This means the following in various settings,

- **Matrix scaling**: A is a non-negative rational $n \times n$ matrix. $\|A\| = \sum_{i,j} A_{i,j}$.
- **Operator scaling**: A is a tuple of $n \times n$ matrices, $A = (A_1, \ldots, A_m)$, with entries in $\mathbb{Q}$. $\|A\| = (\sum_{i=1}^m \|A_i\|_F^2)^{1/2}$.
- **Tensor scaling**: A is a tuple of tensors (in $\mathrm{Ten}(n_1, \ldots, n_d) = \mathbb{C}^{n_1} \otimes \cdots \otimes \mathbb{C}^{n_d}$), $A = (A_1, \ldots, A_m)$, with entries in $\mathbb{Q}$. $\|A\| = (\sum_{i=1}^m \|A_i\|_2^2)^{1/2}$.

Output: Either the algorithm correctly identifies that A is not scalable, or it outputs a scaling A' of A s.t. $\widetilde{\mathrm{ds}}(A') \leqslant \varepsilon$. The measure $\widetilde{\mathrm{ds}}$ is the same as ds for tensor scaling (Definition 2.21) while for matrix and operator scaling, there is a minor variation as explained below,[18]

- **Matrix scaling**: $\widetilde{\mathrm{ds}}(A) = \sum_{i=1}^n (r_i - 1/n)^2 + \sum_{j=1}^n (c_j - 1/n)^2$ (cf. Definition 2.3).
- **Operator scaling**: $\widetilde{\mathrm{ds}}(A) = \left\|\sum_{i=1}^m A_i A_i^\dagger - I_n/n\right\|_F^2 + \left\|\sum_{i=1}^m A_i^\dagger A_i - I_n/n\right\|_F^2$ (cf. Definition 2.10).

Algorithm:

1. Check if some trivial conditions hold. If these do not hold, then output not scalable and halt.
2. There is a group $\widetilde{G}$ and a subset of the group $H \subseteq \widetilde{G}$. In essence H corresponds to the normalization steps in the scaling algorithms from Section 2. We will iteratively act on the current object by elements in H. H will always be the subset corresponding to tuples of positive definite matrices s.t. at most one matrix in the tuple is not identity. For matrix scaling, $\widetilde{G} = \mathrm{T}(n) \times \mathrm{T}(n)$, which acts as in Definition 2.1. For operator scaling, $\widetilde{G} = \mathrm{GL}(n) \times \mathrm{GL}(n)$, which acts as in Definition 2.8. For tensor scaling, $\widetilde{G} = \mathrm{GL}(n_1) \times \cdots \times \mathrm{GL}(n_d)$, which acts as in Definition 2.17.
3. Let $A^{(0)} = A$. For T iterations, $t = 0$ to $T - 1$:
   - If $\widetilde{\mathrm{ds}}\left(A^{(t)}\right) \leqslant \varepsilon$, then output $A^{(t)}$ and halt. Otherwise $A^{(t+1)} = h^{(t)} \cdot A^{(t)}$. $h^{(t)}$ is chosen according to some rule but all we will need for the analysis is that $\|A^{(t)}\| = 1$ is preserved throughout and as a consequence the non-identity element in the tuple of matrices $h^{(t)}$, denoted by $\widehat{h^{(t)}}$, satisfies $\mathrm{tr}\left[\left(\widehat{h^{(t)}}\right)^{-k}\right] = n'$ ($k = 1$ for matrix scaling and $k = 2$ for operator and tensor scaling; we ask the reader to verify this in all the three cases). Also as a consequence of $\widetilde{\mathrm{ds}}\left(A^{(t)}\right) > \varepsilon$, $\mathrm{tr}\left[\left(\left(\widehat{h^{(t)}}\right)^{-k} - I_{n'}\right)^2\right] \geqslant \varepsilon'$. Here n' is the dimension of $h^{(t)}$ and $\varepsilon' = (n')^2\varepsilon/2$ for matrix and operator scaling, and $\varepsilon' = (n')^2\varepsilon/d$ for tensor scaling.
4. Output that A is not scalable.

Algorithm 4: Common template for Algorithms 1, 2 and 3 (has a different normalization).

2. **Progress per step**: As long as $\widetilde{\mathrm{ds}}\left(A^{(t)}\right) \geqslant \varepsilon$,

$$\Phi\left(A^{(t+1)}\right) \geqslant \exp(\varepsilon'/12)\, \Phi\left(A^{(t)}\right)$$

This follows from the invariance property of P (under the action of G) and Eq. (3). Since P is invariant under the action of G, it follows that

$$P(h \cdot A) = \mathrm{Det}\left(\widehat{h}\right)^{\ell/n'} P(A)$$

for all $h \in H$ (here $\widehat{h}$ is the only non-identity matrix in the tuple h).

3. **Upper bound**: $\Phi\left(A^{(t)}\right) \leqslant U$ for all t because of Eq. (2) and due to the fact that $\|A^{(t)}\| = 1$ for all t.

The above three steps imply Theorem 4.1. It is quite magical that these invariant polynomials end up being useful potential functions for the analysis of these scaling algorithms. Without realizing the group actions at play, it might have been quite challenging to come up with potential functions for operator and tensor scaling algorithms (matrix scaling was done in [LSW98] without realizing the invariant theoretic connection). We also remark that most of the previous works use certain optimization problems called capacity (related to Eq. (1)) as potential functions but invariant polynomials lie at the heart of the analysis.

The only thing left to complete the analysis is to get a handle on U in Eq. (2). This is something that needs to be done differently for different scaling problems and we proceed to do this next.

4.1 Potential functions for matrix scaling

Here the polynomials P are extremely simple. We know that A is not in the null cone iff the bipartite graph defined by $\mathrm{supp}(A)$ has a perfect matching (Corollary 3.5 and the succeeding discussion). Hence one can just take $P(A) = \prod_{i=1}^{n} A_{i,\sigma(i)}$ for an appropriate permutation $\sigma \in S_n$. This polynomial satisfies Eq. (2) with $U = 1$. We then get Theorem 2.6 from Theorem 4.1.[20]

4.2 Potential functions for operator scaling

For the group action corresponding to operator scaling, i.e. left-right action (see Section 3.3.2), there is an explicit description of invariants.

Theorem 4.3 ([DW00, DZ01, SdB01, ANS10])**.** *The invariant ring of polynomials of the left-right action is generated by all polynomials of the form* $\mathrm{Det}\left(\sum_{i=1}^{m} D_i \otimes A_i\right)$*, where all* D_i*'s are* $k \times k$ *matrices and* k *varies over* $\mathbb{N}$*.*

This implies the following,

[20]There is a slight discrepancy in parameters. This is because in Algorithm 4, we started already with normalized A's.

Corollary 4.4. *$A = (A_1, \ldots, A_m)$ is scalable iff there exists $k \in \mathbb{N}$ and $k \times k$ matrices D_i's s.t.*

$$P(A) = \mathrm{Det}\left(\sum_{i=1}^{m} D_i \otimes A_i\right) \neq 0$$

Through an appropriate application of Alon's combinatorial nullstellansatz [Alo99], the AM-GM inequality and Cauchy-Schwarz inequality, the following can be proved [GGOW16],

Corollary 4.5. *$A = (A_1, \ldots, A_m)$ is scalable iff there exists $k \in \mathbb{N}$ and $k \times k$ integer matrices D_i's s.t.*

$$P(A) = \mathrm{Det}\left(\sum_{i=1}^{m} D_i \otimes A_i\right) \neq 0$$

and also,

$$|P(A)| \leqslant n^{nk/2}\|A\|^{nk}$$

Since the degree is nk, we get that $U = \sqrt{n}$ in this case.[21] Hence we get Algorithm 2 from Theorem 4.1.

4.3 Potential functions for tensor scaling

For tensor actions an explicit descriptions of the invariants is not known (for examples it is not known if there is a basis of invariant polynomials which are efficiently computable). However a semi-explicit description is known that can be used to bound U by $n_1 n_2 \cdots n_d$. See [BGO$^+$18] for details. Hence one gets Theorem 2.25 from Theorem 4.1.

We want to remark that the bounds on U for operator and tensor scaling cases use sophisticated methods from invariant theory. Somewhat naive methods, e.g. reducing the problem to bounds on solutions to linear systems, only yield bounds on $|P(A)|$ which are doubly exponential in the degree ℓ (as opposed to singly exponential ℓ) which would be useless in the tensor case since we can only bound the required degree by an exponential in the dimensions [Der01]. For the left-right action though, ℓ can be assumed to be at most $n - 1$ due to the work of Derksen and Makam [DM15].

5 Applications of scaling

As we saw in the previous chapters, scaling problems have a surprising connection to invariant theory, which turns out to be fundamental to the analysis of the alternate minimization algorithms which solve the scaling problems. This connection naturally leads to new and efficient algorithms for problems in invariant theory. In this section we will see even more applications of scaling problems in different areas of science. For a complete discussion of the applications of matrix, operator and tensor scaling, we refer the reader to the papers [SZ90, Ide16, GGOW16, GGOW17, BGO$^+$18].

[21] While a similar bound on U follows from the discussion for tensor scaling (after all operator scaling is a special case), the method described here is more explicit.

5.1 Matrix Scaling

The matrix scaling problem has been posed (sometimes independently) in many different areas of study, ranging from telephone forecasting [Kru37], economics [Sto62], statistics [Sin64], image reconstruction algorithms [HL76], linear algebra [FLS88, BDYW11], optimization [RS89] and theoretical computer science [LSW98, BDYW11]. For a more comprehensive list of references and historical overview of scaling problems, we recommend the survey [Ide16], the papers [RS89, SZ90] and references therein. In this section we will describe three of the applications of matrix scaling cited above, providing a peek on the abundance of applications of this simple and natural problem.

Computer Science: Given a non-negative matrix $A \in \mathrm{Mat}_n(\mathbb{R})$, its permanent is given by the following expression:

$$\mathrm{perm}(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} A_{i,\sigma(i)},$$

where S_n is the group of permutations of the set $\{1, 2, \ldots, n\}$. This polynomial is extremely important in computer science, due to its completeness for a number of complexity classes.

Computing the permanent of a 0-1 matrix is a #P-complete problem, as shown by Valiant in [Val79b]. Thus, much research has been devoted to computing a multiplicative approximation to the permanent, as described in [LSW98] and references therein.

Given a matrix $A \in \mathrm{Mat}_n(\mathbb{R})$, note that the permanent of any scaling BAC of A is given by

$$\mathrm{perm}(BAC) = \prod_{i=1}^{n} B_{i,i} C_{i,i} \cdot \mathrm{perm}(A).$$

Thus, if we could find diagonal matrices B, C for which we knew a good approximation for $\mathrm{perm}(BAC)$, the equality above would give us a good approximation for $\mathrm{perm}(A)$. As it turns out, for doubly-stochastic matrices, good lower and upper bounds on the permanent are known! If D is a doubly-stochastic matrix, the upper bound $\mathrm{perm}(D) \leqslant 1$ is trivial. For the lower bound, the solution to van der Waerden's conjecture gives us that $\mathrm{perm}(D) \geqslant \frac{n!}{n^n} \geqslant e^{-n}$ [Fri79, Fal81, Ego81, Gur04b, Gur08]. Thus, finding a scaling of A to a doubly stochastic matrix gives us an e^n approximation to compute the permanent! This approximation was given in the work of Linial et al. [LSW98]. While a fully polynomial randomized approximation scheme (FPRAS) is known for approximating the permanent [JSV04], the above algorithm, despite giving a much worse approximation (still a non-trivial one), is deterministic. The current best deterministic algorithm for approximating the permanent is due to Gurvits and Samorodnitsky [GS14] (their paper also uses matrix scaling!), which achieves an approximation factor of 2^n.

Combinatorial Geometry: The Sylvester-Gallai theorem states that if m distinct points $p_1, \ldots, p_m \in \mathbb{R}^n$ are arranged such that for any two distinct points p_i, p_j, there exists a third point p_k on the line defined by p_i and p_j, then it must be the case that all points are collinear, that is, lie in a 1-dimensional subspace of $\mathbb{R}^n$. This basic theorem in

combinatorial geometry has many variants and generalizations, which can be found in [BM90, BDYW11, DGOS16] and references therein.

A more quantitative version of this problem, known as the δ-SG problem, is defined as follows: if we now assume that the m distinct points (could take them in $\mathbb{C}^n$) are arranged such that for any point p_i, there are at least δm points p_j such that the line through p_i and p_j contains a third point p_k, can we say that these points lie in a low dimensional subspace? Note that the original Sylvester-Gallai theorem is a special case when $\delta = 1$. As it turns out, this δ-SG problem can be phrased as a problem in linear algebra: given a matrix $P \in \mathbb{C}^{m\times n}$ whose rows are given by the points $p_1, \ldots, p_m$ satisfying the arrangement contraints, is P a low rank matrix?

One approach to prove that such a matrix P is low rank is to find a high rank matrix $A \in \mathbb{C}^{\ell\times m}$ such that $AP = 0$. In the case of the δ-SG problem, a natural matrix suggests itself: take A to be the matrix which characterizes the dependencies of the points $p_1, \ldots, p_m$. That is, for each triple (i, j, k) such that p_i, p_j, p_k are collinear, simply add a row to A which encodes the linear combination of p_i, p_j, p_k which gives zero. As it will soon be clear, this matrix A which arises in the δ-SG problem is a very special type of matrix, and matrix scaling helps prove that such matrices are always of high rank.

A surprising application of matrix scaling arises when one tries to obtain lower bounds on the rank of special types of matrices, called *design matrices*, which are defined based on the pattern of zero/non-zero entries in the matrix. More precisely, we say that a matrix $B \in \mathbb{C}^{\ell\times m}$ is a (q, k, t)-design matrix if each row of B has at most q non-zero entries, each column has at least k non-zero entries, and the supports of any two columns intersect in at most t rows. Note that the matrix A from the δ-SG problem is an example of a design matrix. In [BDYW11], the authors used matrix scaling to prove that any $\ell \times m$ matrix, where $\ell \geqslant m$, which is a (q, k, t)-design matrix has rank at least $m - \left(\frac{qtm}{2k}\right)^2$. With this bound, they proved that any δ-SG configuration must be in a subspace of dimention at most $13/\delta^2$. This bound was improved in [DSW14] to $12/\delta$ by giving better bounds on ranks of design matrices (again relying heavily on matrix scaling).

Statistics: It turns out that matrix scaling has an equivalent formulation as an entropy optimization problem, thereby being very useful in statistics.

Problem 5.1 (Matrix Scaling - entropy formulation)**.** *Given two probability distributions $r, c \in \mathbb{R}^n$ and a non-negative matrix $A \in \mathrm{Mat}_n(\mathbb{R})$, find a non-negative matrix B^* s.t.* [22]

$$B^* = \text{argmin } D(B||A) \text{ s.t.}$$
$$\sum_{j=1}^{n} B_{ij} = r_i, \forall i \in [n]$$
$$\sum_{i=1}^{n} B_{ij} = c_j, \forall j \in [n]$$

[22] $D(B||A) = \sum_{i,j} B_{i,j} \log(B_{i,j}/A_{i,j})$ is the KL-divergence between B and A.

It turns out that the above optimization problem is equivalent to a non-uniform version of the matrix scaling problem, where one wants to scale A to achieve marginals r, c (as opposed to all 1's). The above optimization problem is trying to recover a joint probability distribution B^* based on knowledge of the marginals r, c (w.r.t. to some initial distribution A). Such estimation of joint probability distributions from partial data is abundant in statistics, as pointed out by [SZ90], with examples coming from estimating contingency tables, interregional migration, deriving probability estimates from census data, and many others.

5.2 Operator scaling

Non-Commutative Algebra: Whenever a certain mathematical object can be represented in several equivalent ways, a natural question which arises is the so called *word problem*: given two representations of a mathematical object, do they describe the same object? Word problems are fundamental across many subareas of mathematics. In non-commutative algebra, when defining the *free skew field*, which is the field given by all rational functions over non-commutative variables (that is, the non-commutative equivalent of the field of rational functions), the word problem arises as a computational problem in a very natural way, which we describe next.

By the foundational work of Amitsur [Ami66], the elements of the free skew field can be described by equivalence classes of arithmetic formulas, which take non-commutative variables and elements of the base field as inputs and use linear combinations, multiplications and inverse gates to compute non-commutative rational expressions. Two rational expressions are said to be equivalent if they have the same evaluation[23] when we substitute the non-commuting variables by $d \times d$ matrices, for every $d \in \mathbb{N}$.

Therefore, we can phrase the word problem for the free skew field in a natural computational way: given two non-commutative arithmetic formulas with inversion gates, are they equivalent (i.e. do they compute the same rational function)? Note that Amitsur's work still leaves open the decidability of the word problem for the free skew field, as he provided no bounds on the dimension d. To prove that this word problem is decidable, in a series of works [Coh71, Coh73, Coh75], Cohn reduced the word problem above to the problem of *non-commutative singularity testing:* given a linear symbolic matrix $L = A_1x_1 + \cdots + A_mx_m$, where $A_i \in \mathrm{Mat}_n(\mathbb{F})$ and x_i's are non-commutative variables, is the matrix L singular in the free skew field? In this series of works, Cohn essentially proved that any non-commutative formula can be computed by the (1,1) entry of the inverse of such a symbolic matrix, which is the non-commutative analog of Valiant's completeness of determinant for commutative formulas [Val79a].

The connection between the non-commutative singularity problem and operator scaling comes from a theorem of Cohn [Coh95] which establishes that the symbolic matrix L above is singular over the free skew field if, and only if, the tuple of matrices $(A_1, \ldots, A_m)$ is dimension non-decreasing (see Definition 2.12). Therefore, by Theorem 2.13, given a symbolic matrix, to test its singularity we only need to test whether the tuple $(A_1, \ldots, A_m)$ can be scaled to doubly-stochastic![24]

[23] whenever they are defined in the given inputs, that is, the formula does not invert a singular matrix.

[24] Cohn's reduction works over any field $\mathbb{F}$. However, operator scaling only makes sense over $\mathbb{Q}, \mathbb{R}, \mathbb{C}$, and therefore our algorithm only works when the base field is one of the three fields just mentioned. A

Invariant Theory: As we saw in Section 3, a fundamental problem in computational invariant theory is the *null-cone problem*: given a group G acting on a vector space V, and a point $v \in V$, decide whether v is in the null cone $\mathcal{N}_G(V)$.

As it was also discussed in Section 3, the operator scaling problem corresponds to the null-cone problem for the left-right action. Thus, the scaling algorithm and its analysis prove that for this particular action the null-cone problem is in P. This was the first polynomial time algorithm for the null-cone problem for the left-right action.

Combinatorial Geometry: As we have seen in Section 5.1, matrix scaling is very useful to prove rank bounds for design matrices, and such bounds found applications in combinatorial geometry. In very recent work [DGOS16], the authors generalize the definition of a design matrix to block matrices, which we will soon define, and used operator scaling to prove rank bounds for design block matrices.

These bounds were then used to obtain three new applications in combinatorial geometry: bounding the projective rigidity of a configuration of points, obtaining tight bounds on a generalization of the quantitative Sylvester-Gallai problem seen in Section 5.1 and upper bounding the dimension of spaces containing certain configurations of low degree curves with many incidences, which can be seen as a variant of the Sylvester-Gallai theorem to low degree curves.

We say that a matrix A is an $\ell \times m$ block matrix, with $d \times d$ blocks, if A is a matrix whose entries A_{ij} are matrices of dimension $d \times d$. When $d = 1$, we obtain the usual definition of an $\ell \times m$ matrix. For a block matrix A, we denote its rank to be the rank of the $\ell d \times md$ matrix $\tilde{A}$ obtained from A by ignoring the block structure. With this definition in mind, we can define *design block matrices* simply as follows: A is a (q, k, t)-design block matrix if each row of A has at most q non-zero blocks, each column of A contains at least k non-singular blocks and for any two columns, their support intersects in at most t rows. In [DGOS16], the authors obtain rank bounds for block design matrices, generalizing the results obtained for design matrices and discussed in Section 5.1. Operator scaling played a crucial rule in the proof. Note that a design block matrix cannot be thought of as a design matrix with some small blowup in parameters, since the columns corresponding to the same block could intersect at a lot of places, and hence the bounds of [BDYW11, DSW14] are not applicable in this setting.

5.3 Brascamp-Lieb inequalities and polytopes

Another important application of operator scaling appears in functional analysis and optimization, towards the celebrated *Brascamp-Lieb inequalities* [BL76, Lie90] and their corresponding *Brascamp-Lieb polytopes*. The Brascamp-Lieb inequalities (BL for short) and their reverse form generalize many important inequalities, such as Cauchy-Schwarz and Hölder's inequalities, Loomis-Whitney inequality, Young's convolution inequalities and many others. In this section, we will describe how BL-inequalities can be seen as a particular case of the operator scaling problem, and discuss some applications in combinatorics and complexity. For a more in depth discussion of BL inequalities, we refer the reader to the papers [BCCT08, GGOW17] and references therein.

completely algebraic algorithm given by Ivanyos et al. [IQS17b, IQS17a] solves the singularity problem over any field.

A BL datum is given by a tuple of matrices $\mathbf{B} = (B_1, \ldots, B_m)$ where $B_i \in \mathbb{R}^{n_i \times n}$ and a tuple of non-negative reals $\mathbf{p} = (p_1, \ldots, p_m)$. We will represent a BL datum by the tuple $(\mathbf{B}, \mathbf{p})$. The BL inequality with datum $(\mathbf{B}, \mathbf{p})$ states that for every tuple of non-negative, Lebesgue integrable functions $(f_1, \ldots, f_m)$, where $f_i : \mathbb{R}^{n_i} \to \mathbb{R}$, the following inequality holds:[25]

$$\int_{x \in \mathbb{R}^n} \prod_{i=1}^{m} f_i(B_i x) dx \leqslant C \cdot \prod_{i=1}^{m} \|f_i\|_{1/p_i}, \tag{4}$$

for some constant $C \in (0, \infty]$ which is independent of the functions f_i. When C is finite (which is when we indeed have a non-trivial inequality), we say that the datum $(\mathbf{B}, \mathbf{p})$ is feasible, and denote by $BL(\mathbf{B}, \mathbf{p})$ the smallest value of C for which the inequality always holds, which we refer to as the BL constant.

Ball [Bal89] and Barthe [Bar98] proved that for certain types of BL data, the BL constant will always equal 1. They called these types of BL data *geometric*, which we now define:

Definition 5.2 (Geometric BL datum)**.** *A BL datum is called* geometric *if it satisfies the following conditions:*

1. ***Isotropy:*** $\sum_{i=1}^{m} p_i B_i^T B_i = I_n$.
2. ***Projection:*** *For every* $i \in [m]$, B_i *is a projection matrix, that is,* $B_i B_i^T = I_{n_i}$.

Note that the definition above is remarkably similar to the definition of doubly stochastic operators (Definition 2.9), and this similarity is an important part of the connection between BL inequalities and operator scaling.

An important characterization of feasible BL data was given by Bennet et al. [BCCT08], and this characterization, as we will soon see, is remarkably similar to the dimension non-decreasing definition (Definition 2.12).

Theorem 5.3 (Feasibility of BL datum [BCCT08])**.** *The datum* $(\mathbf{B}, \mathbf{p})$ *is feasible iff the following inequalities hold:*

1. $n = \sum_{i=1}^{m} p_i n_i$
2. $\dim(V) \leqslant \sum_{i=1}^{m} p_i \dim(B_i(V))$, *for all subspaces* $V \subseteq \mathbb{R}^n$.

Note that for a given tuple of matrices $\mathbf{B}$, the theorem above proves that the set of vectors $\mathbf{p}$ for which the datum $(\mathbf{B}, \mathbf{p})$ is feasible is given by a polytope.[26] This polytope, which we denote by $P_{\mathbf{B}}$, is the so called BL polytope.

In [GGOW17], the authors reduce the feasibility of BL data to an operator scaling problem, thereby giving an efficient algorithm for the membership problem in a BL polytope (that is, solving the problem: given datum $(\mathbf{B}, \mathbf{p})$, is $\mathbf{p} \in P_{\mathbf{B}}$?). We refer the reader to [GGOW17] for details (or perhaps find the reduction yourself!).

The reduction from BL to operator scaling along with Theorem 2.13 yields the following BL scaling theorem (which is already present in the work of [BCCT08]).

[25] Below, $\|f_i\|_{1/p_i} = \left(\int_{y_i \in \mathbb{R}^{n_i}} f_i(y_i)^{1/p_i} dy_i\right)^{p_i}$.

[26] One can see that the number of inequalities is finite because given a tuple $\mathbf{B}$, the numbers $\dim(B_i V)$ lie in the set $[n]$, therefore giving us at most n^m different inequalities.

Theorem 5.4. *A BL datum* $(\mathbf{B}, \mathbf{p})$ *is feasible iff there are invertible matrices (denoted* BL scalings*)* $A \in \mathrm{GL}_n(\mathbb{R})$, $C_i \in \mathrm{GL}_{n_i}(\mathbb{R})$ *such that the datum* $(\mathbf{B}', \mathbf{p})$ *is geometric, where* $\mathbf{B}' = (C_1 B_1 A, \ldots, C_m B_m A)$.[27]

With the definitions and results above, we are ready to discuss some applications of BL inequalities and polytopes.

Complexity Theory: Forster's celebrated lower bound on the unbounded error probabilistic communication complexity or the sign rank uses a remarkable result, proved by him in [For02, Theorem 4.1], which can be stated as follows:

Theorem 5.5 ([For02])**.** *Let* $v_1, \ldots, v_m \in \mathbb{R}^n$*, where* $m \geqslant n$*, be a set of vectors in general position, that is, any subset of* n *of these vectors are linearly independent. Then, there exists a matrix* $A \in \mathrm{GL}_n(\mathbb{R})$ *such that the following holds:*

$$\sum_{i=1}^{m} \frac{n}{m} \cdot \frac{(Av_i)(Av_i)^T}{\|Av_i\|_2^2} = I_n \tag{5}$$

The condition given above, when stated in the language of BL inequalities, becomes exactly the claim that the BL datum $(\mathbf{B}, \mathbf{p})$ given by $B_i = v_i^T$ and $p_i = n/m$ for $i \in [m]$, can be scaled to a geometric datum when the vectors v_i are in general position! We note that generalizations of Forster's theorem already appear in two previous works, explicitly in [GS02] and implicitly in [Bar98].

Since Forster's result is a special case of Theorem 5.4, which itself is a special case of the operator scaling problem, we see that the operator scaling theory gives a vast generalization to Forster's theorem.

Combinatorial Optimization: BL polytopes are interesting combinatorially because they can be very complex, having exponentially many facets, while admitting a very succinct description (given by the tuple of matrices $\mathbf{B}$). The work in [GGOW17] gives a membership oracle, as well as a separation oracle for BL polytopes[28] and therefore these polytopes could be a useful tool for solving natural optimization problems. Thus, looking for natural polytopes which are special cases of BL polytopes is a first step in understanding their expressive power, which is far from understood.

One polytope which can be encoded as a BL polytope in a simple way is the linear matroid intersection polytope, which we now describe. The linear matroid associated with a tuple of vectors $v = (v_1, \ldots, v_m)$, where $v_i \in \mathbb{R}^n$, is the matroid with the following collection of independent sets: $\mathcal{M}_v = \{I \subseteq [m] \mid (v_i)_{i \in I} \text{ are linearly independent}\}$.

Given two tuples of vectors $v = (v_1, \ldots, v_m)$ and $w = (w_1, \ldots, w_m)$, defining two linear matroids $\mathcal{M}_v, \mathcal{M}_w$ over $\mathbb{R}^n$, their (linear matroid) intersection polytope is given by the convex hull of the characteristic vectors of their common independent sets. That is,

$$P_{\mathcal{M}_v, \mathcal{M}_w} = \mathrm{conv}\{1_I \mid I \subseteq [m] \text{ s.t. } (v_i)_{i \in I} \text{ and } (w_i)_{i \in I} \text{ are linearly independent}\}.$$

[27] The precise statement would be that the new datum $(\mathbf{B}', \mathbf{p})$ is "*close to* geometric," which is analogous to what happens in the operator scaling setting. For simplicity, we forgo the exact statement.

[28] The running time of the oracles depends on the common denominator of the vector $\mathbf{p}$. For more details, we refer the reader to the paper.

In [GGOW17], the authors prove[29] that the polytope $P_{\mathcal{M}_v,\mathcal{M}_w}$ corresponds to the BL polytope given by the matrices $B_i = \begin{pmatrix} 0 & v_i^T \\ w_i^T & 0 \end{pmatrix}$, where each 0 corresponds to the zero vector in $\mathbb{R}^n$.

5.4 Tensor scaling

Entanglement distillation: The tensor scaling problem has a very natural interpretation in quantum information theory. If we regard the vector space $V = \mathrm{Ten}(n_1, \ldots, n_d)$ as the set of pure states of a quantum system with d particles,[30] the scaling action of $G = \mathrm{SL}(n_1) \otimes \cdots \otimes \mathrm{SL}(n_d)$ corresponds to a class of quantum operations called *stochastic local operations and classical communication (SLOCC)*, defined in [BPR$^+$00]. These operations on a quantum system have a natural communication complexity interpretation: each party is holding a particle of the system, parties are allowed free classical communication (i.e. sending bits to one another) and each party can perform quantum operations and measurements on its own particle, and finally we allow *post selection* on measurement outcomes.

Quantum states with uniform marginals are called *locally maximally entangled* and hence the tensor scaling question is about distilling locally maximally entangled states from a given pure state by SLOCC operations.

Slice-Rank: The slice-rank of a tensor, introduced in [Tao16], is a different notion of tensor rank which has found applications in extremal combinatorics and number theory (for more details, see [BCC$^+$17] and references therein). A tensor $B \in \mathrm{Ten}(n_1, \ldots, n_d)$ is said to have *slice-rank one* if it is the tensor product of a vector and a lower order tensor, that is, if there exists an index $j \in [d]$ such that $B = v \otimes_j C$, where $v \in \mathbb{C}^{n_j}$ and $C \in \mathrm{Ten}(n_1, \ldots, n_{j-1}, n_{j+1}, \ldots, n_d)$. With this definition, the *slice-rank* of a tensor, $A \in \mathrm{Ten}(n_1, \ldots, n_d)$, is defined as the smallest k for which A can be decomposed as the sum of k slice-rank one tensors. We denote the slice rank of A by $\text{slice-rank}(A)$.

Let us focus on the case $n_1 = \cdots = n_d = n$ (not all of what we are going to say holds in the unbalanced case). Note that the slice-rank of a tensor can be at most n, since we can always flatten the tensor on any one coordinate, and the matrix rank decomposition of the flattening provides a slice decomposition of the tensor. In a recent work [BCC$^+$17], the authors developed a connection between the slice rank of a tensor (and an asymptotic version of slice-rank) and the null cone of the tensor scaling group action. More precisely the authors proved the following two theorems:

Theorem 5.6. *Given $A \in \mathrm{Ten}(n, \ldots, n)$, if $\text{slice-rank}(A) < n$, then A is in the null cone of the tensor scaling action.*

Theorem 5.7. *If a tensor $A \in \mathrm{Ten}(n, \ldots, n)$ is in the null cone of the tensor action, then there exists $k \in \mathbb{N}$ such that $\text{slice-rank}(A^{\otimes k}) < n^k$.*

[29] There was a mistake in their original proof, which was fixed thanks to Damien Strazak and Nisheeth Vishnoi.

[30] We would have to consider only tensors of unit norm.

Therefore deciding whether a particular tensor lies in the null cone of the tensor action could give us information on its slice-rank or on the asymptotic version of slice-rank. Relying on Theorems 5.6 and 5.7, it was proved in [BGO$^+$18] that being in the null cone of the tensor action and non-fullness of the asymptotic slice-rank are equivalent conditions (in other words Theorem 5.7 is in fact an equivalence).

6 Conclusion and open problems

We have seen that scaling problems are particular instances of fundamental problems in invariant theory, and that invariant theory provides a rich source of potential functions to analyze the natural alternating minimization algorithms for the scaling problems. Therefore, settling the complexity of problems in invariant theory will have many applications not only in computational invariant theory or geometric complexity theory, but also in many other areas of science. We believe that the recent series of works on scaling algorithms are only the beginning of many future discoveries, and to witness this we present several problems which are still open in the area:

1. Design a polynomial time algorithm for tensor scaling with a $\mathrm{poly}(\log(1/\varepsilon))$ dependence on the error parameter ε. This will yield a polynomial time algorithm for the null-cone problem for tensor actions. Such algorithms already exist for the matrix scaling [KK96, LSW98, CMTV17, ALOW17] and the operator scaling [AZGL$^+$18] problems. This involves exploring algorithms for geodesically convex optimization (see [AZGL$^+$18] and references therein for a discussion).

2. Is there a polynomial time algorithm for the null-cone problem for more general group actions? The moment map and the optimization problem in Eq. (1) provide an analytic approach to this.

3. Can one understand the behavior of Algorithms 1, 2 and 3 when the object A is not scalable? Does the algorithm converge to some cycle? How close does one get to satisfying the stochasticity constraints?

4. Can we find more applications of scaling problems in computer science? More generally, can we find instances of the non-commutative duality appearing in computer science?

5. As mentioned in Section 4, the analysis of the scaling algorithms rely on the existence of a generating set of invariant polynomials which is "nice" in the following sense: each polynomial in this generating set has "small" coefficients (of size exponential in the degree of the polynomials). Is there a systematic way of obtaining such nice set of polynomials for more general group actions?

7 Other recent scaling works

This survey covered a particular family of scaling problems, which is usually referred to as *uniform scaling problems*, as the matrix, operator and tensor scaling problems here

defined only concern with the possibility of scaling the input to a doubly-stochastic (matrix and operator scaling) or a d-stochastic element (tensor scaling). However, this is not the whole story, as one could ask the following question: given prescribed marginals and an input matrix/operator/tensor, can we scale the input to have the prescribed marginals?

This more general question has also been studied extensively. In the matrix scaling case, the theory of non-uniform scaling is not much different from the theory of uniform scaling. However in the operator and tensor scaling settings, there are a lot more twists in the non-uniform case. Recent works have made remarkable progress towards this end [Fra18, BFG$^+$18]. In this more general setting, more sophisticated concepts from invariant theory and representation theory are needed for the analysis of the algorithms.

Another line of research has been in the development of faster algorithms for scaling problems, with a different approach than the alternating minimization described in this survey. Recently some remarkable successes have been obtained in this direction, with nearly-linear algorithms being developed for matrix scaling [CMTV17, ALOW17] and a faster algorithm ($\mathrm{poly}(\log(1/\varepsilon))$ convergence rate as opposed to $\mathrm{poly}(1/\epsilon)$) being developed for the operator scaling problem [AZGL$^+$18].

Other variants of the scaling problems above have also been studied, and we cite here a few. The *matrix balancing* problem is a variant of matrix scaling where given a square matrix A with complex entries, the goal is to decide whether there is a diagonal matrix D such that DAD^{-1} is *doubly-balanced*, that is, where the i^{th} row has the same norm as the i^{th} column. This problem also has applications in different areas of computer science and numerical analysis. For more details on matrix balancing, we refer the reader to the survey [Ide16] and references therein. In [KLLR17], the authors solve the Paulsen problem in operator theory and they use several tools from the theory of operator scaling, in addition to other sophisticated methods.

References

[AFG14] Michael A. Forbes and Venkatesan Guruswami. Dimension expanders via rank condensers. *arXiv preprint arXiv:1411.7455*, 2014.

[Alo99] Noga Alon. Combinatorial Nullstellensatz. *Combinatorics, Probability and Computing*, 8(1-2):7–29, 1999.

[ALOW17] Zeyuan Allen-Zhu, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Much faster algorithms for matrix scaling. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 890–901, 2017. URL: `https://doi.org/10.1109/FOCS.2017.87`, `doi:10.1109/FOCS.2017.87`.

[Ami66] Shimshon Amitsur. Rational identities and applications to algebra and geometry. *Journal of Algebra*, 3:304–359, 1966.

[ANS10] Bharat Adsul, Suresh Nayak, and K. V. Subrahmanyam. A geometric approach to the kronecker problem ii : rectangular shapes, invariants of

matrices, and a generalization of the Artin-Procesi theorem. *Manuscript, available at http://www.cmi.ac.in/ kv/ANS10.pdf*, 2010.

[AZGL+18] Zeyuan Allen-Zhu, Ankit Garg, Yuanzhi Li, Rafael Oliveira, and Avi Wigderson. Operator scaling via geodesically convex optimization, invariant theory and polynomial identity testing. *STOC*, 2018.

[Bal89] Keith Ball. Volumes of sections of cubes and related problems. *Geometric Aspects of Functional Analysis*, pages 251–260, 1989.

[Bar98] Franck Barthe. On a reverse form of the Brascamp–Lieb inequality. *Inventiones Mathematicae*, 134:335–361, 1998.

[BCC+17] Jonah Blasiak, Thomas Church, Henry Cohn, Joshua A. Grochow, Eric Naslund, William F Sawin, and Chris Umans. On cap sets and the group-theoretic approach to matrix multiplication. *Discrete Analysis*, 2017. URL: `http://arxiv.org/abs/1605.06702`, `arXiv:1605.06702`, `doi:10.19086/da.1245`.

[BCCT08] Jonathan Bennett, Anthony Carbery, Michael Christ, and Terence Tao. The Brascamp-Lieb inequalities: finiteness, structure, and extremals. *Geometric and Functional Analysis*, 17(5):1343–1415, 2008.

[BDYW11] Boaz Barak, Zeev Dvir, Amir Yehudayoff, and Avi Wigderson. Rank bounds for design matrices with applications to combinatorial geometry and locally correctable codes. In *Proceedings of the forty-third annual ACM symposium on Theory of computing*, pages 519–528. ACM, 2011.

[BFG+18] Peter Bürgisser, Cole Franks, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Efficient algorithms for tensor scaling, quantum marginals and moment polytopes. *arXiv preprint arXiv:1804.04739*, 2018.

[BGO+18] Peter Bürgisser, Ankit Garg, Rafael Oliveira, Michael Walter, and Avi Wigderson. Alternating minimization, scaling algorithms, and the null-cone problem from invariant theory. *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, 2018. `arXiv:1711.08039`.

[BI11] Peter Bürgisser and Christian Ikenmeyer. Geometric complexity theory and tensor rank. In *Proceedings of the Symposium on the Theory of Computing (STOC 2011)*, pages 509–518. ACM, 2011. `arXiv:1011.1350`, `doi:10.1145/1993636.1993704`.

[BI13] Peter Bürgisser and Christian Ikenmeyer. Explicit lower bounds via geometric complexity theory. In *Proceedings of the Symposium on the Theory of Computing (STOC 2013)*, pages 141–150. ACM, 2013. `arXiv:1210.8368`, `doi:10.1145/2488608.2488627`.

[BL76] Herm Brascamp and Elliot Lieb. Best constants in Young's inequality, its converse and its generalization to more than three functions. *Advances in Mathematics*, 20:151–172, 1976.

[BM90] Peter Borwein and William OJ Moser. A survey of sylvester's problem and its generalizations. *Aequationes Mathematicae*, 40(1):111–135, 1990.

[BPR+00] Charles H Bennett, Sandu Popescu, Daniel Rohrlich, John A Smolin, and Ashish V Thapliyal. Exact and asymptotic measures of multipartite pure-state entanglement. *Physical Review A*, 63(1):012307, 2000.

[Bür12] Peter Bürgisser. Prospects for geometric complexity theory. In *2012 IEEE 27th Conference on Computational Complexity*, pages 235–235, June 2012. doi:10.1109/CCC.2012.19.

[CMTV17] Michael B. Cohen, Aleksander Madry, Dimitris Tsipras, and Adrian Vladu. Matrix scaling and balancing via box constrained newton's method and interior point methods. In *58th IEEE Annual Symposium on Foundations of Computer Science, FOCS 2017, Berkeley, CA, USA, October 15-17, 2017*, pages 902–913, 2017. URL: https://doi.org/10.1109/FOCS.2017.88, doi:10.1109/FOCS.2017.88.

[Coh71] P. M. Cohn. The embedding of firs in skew fields. *Proceedings of the London Mathematical Society*, 23:193–213, 1971.

[Coh73] P. M. Cohn. The word problem for free fields. *The Journal of Symbolic Logic*, 38(2):309–314, 1973.

[Coh75] P. M. Cohn. The word problem for free fields: A correction and an addendum. *Journal of Symbolic Logic*, 40(1):69–74, 1975.

[Coh95] Paul Cohn. *Skew Fields, Theory of General Division Rings*, volume 57. Encyclopedia of Mathematics and its Applications, 1995.

[Der01] Harm Derksen. Polynomial bounds for rings of invariants. *Proceedings of the American Mathematical Society*, 129(4):955–963, 2001. doi:10.1090/S0002-9939-00-05698-7.

[DGOS16] Zeev Dvir, Ankit Garg, Rafael Oliveira, and József Solymosi. Rank bounds for design matrices with block entries and geometric applications. *arXiv preprint arXiv:1610.08923*, 2016.

[DK15] Harm Derksen and Gregor Kemper. *Computational invariant theory*. Springer, 2015.

[DM15] Harm Derksen and Visu Makam. Polynomial degree bounds for matrix semi-invariants. 2015. arXiv:1512.03393.

[DSW14] Zeev Dvir, Shubhangi Saraf, and Avi Wigderson. Improved rank bounds for design matrices and a new proof of Kelly's theorem. In *Forum of Mathematics, Sigma*, volume 2. Cambridge University Press, 2014.

[DW00] Harm Derksen and Jerzy Weyman. Semi-invariants of quivers and saturation for littlewood-richardson coefficients. *Journal of the American Mathematical Society*, 13(3):467–479, 2000.

[DZ01] Mátyás Domokos and A. N. Zubkov. Semi-invariants of quivers as determinants. *Transformation Groups*, 6(1):9–24, 2001.

[Ego81] Gregory P Egorychev. The solution of van der waerden's problem for permanents. *Advances in Mathematics*, 42(3):299–305, 1981.

[Fal81] Dmitry I Falikman. Proof of the van der waerden conjecture regarding the permanent of a doubly stochastic matrix. *Mathematical notes of the Academy of Sciences of the USSR*, 29(6):475–479, 1981.

[FL89] Joel Franklin and Jens Lorenz. On the scaling of multidimensional matrices. *Linear Algebra and its Applications*, 114-115:717 – 735, 1989. Special Issue Dedicated to Alan J. Hoffman. URL: http://www.sciencedirect.com/science/article/pii/0024379589904904, doi:https://doi.org/10.1016/0024-3795(89)90490-4.

[FLS88] Shmuel Friedland, Chi-Kwong Li, and Hans Schneider. Additive decomposition of nonnegative matrices with applications to permanents and scalingt. *Linear and Multilinear Algebra*, 23(1):63–78, 1988.

[For02] Jürgen Forster. A linear lower bound on the unbounded error probabilistic communication complexity. *Journal of Computer and System Sciences*, 65(4):612–625, 2002.

[Fra18] Cole Franks. Operator scaling with specified marginals. *STOC*, 2018. arXiv:1801.01412.

[Fri79] Shmuel Friedland. A lower bound for the permanent of a doubly stochastic matrix. *Annals of Mathematics*, pages 167–176, 1979.

[GGOW16] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. A deterministic polynomial time algorithm for non-commutative rational identity testing. In *Proceedings of the Symposium on Foundations of Computer Science (FOCS 2016)*, pages 109–117. IEEE, 2016. arXiv:1511.03730, doi:10.1109/FOCS.2016.95.

[GGOW17] Ankit Garg, Leonid Gurvits, Rafael Oliveira, and Avi Wigderson. Algorithmic and optimization aspects of brascamp-lieb inequalities, via operator scaling. In *Proceedings of the Symposium on the Theory of Computing (STOC 2017)*, pages 397–409. ACM, 2017. arXiv:1607.06711.

[GRS13] Valentina Georgoulas, Joel W Robbin, and Dietmar A Salamon. The moment-weight inequality and the Hilbert-Mumford criterion. *arXiv:1311.0410*, 2013.

[GS02] Leonid Gurvits and Alex Samorodnitsky. A deterministic algorithm for approximating the mixed discriminant and mixed volume, and a combinatorial corollary. *Discrete & Computational Geometry*, 27(4):531–550, 2002.

[GS14] Leonid Gurvits and Alex Samorodnitsky. Bounds on the permanent and some applications. In *Foundations of Computer Science (FOCS), 2014 IEEE 55th Annual Symposium on*, pages 90–99. IEEE, 2014.

[Gur04a] Leonid Gurvits. Classical complexity and quantum entanglement. *Journal of Computer and System Sciences*, 69(3):448–484, 2004. doi:10.1016/j.jcss.2004.06.003.

[Gur04b] Leonid Gurvits. Van der waerden conjecture for mixed discriminants. *arXiv preprint math/0406420*, 2004.

[Gur08] Leonid Gurvits. Van der waerden/schrijver-valiant like conjectures and stable (aka hyperbolic) homogeneous polynomials: one theorem for all. *the electronic journal of combinatorics*, 15(1):66, 2008.

[GY98] Leonid Gurvits and Peter N. Yianilos. The deflation-inflation method for certain semidefinite programming and maximum determinant completion problems. *Technical Report, NECI*, 1998.

[HH12] Gert Heckman and Peter Hochs. Geometry of the momentum map, 2012. URL: http://www.maths.adelaide.edu.au/peter.hochs/momentum_new.pdf.

[Hil90] David Hilbert. Ueber die Theorie der algebraischen Formen. *Mathematische Annalen*, 36(4):473–534, 1890.

[Hil93] David Hilbert. Uber die vollen Invariantensysteme. *Math. Ann.*, 42:313–370, 1893.

[HL76] Gabor T Herman and Arnold Lent. Iterative reconstruction algorithms. *Computers in biology and medicine*, 6(4):273–294, 1976.

[Ide16] Martin Idel. A review of matrix scaling and sinkhorn's normal form for matrices and positive maps. *arXiv preprint arXiv:1609.06349*, 2016.

[IQS17a] Gábor Ivanyos, Youming Qiao, and KV Subrahmanyam. Constructive non-commutative rank computation is in deterministic polynomial time. *Proceedings of Innovations in Theoretical Computer Science (ITCS)*, 2017. arXiv:1512.03531, doi:10.4230/LIPIcs.ITCS.2017.55.

[IQS17b] Gábor Ivanyos, Youming Qiao, and KV Subrahmanyam. Non-commutative Edmonds' problem and matrix semi-invariants. *Computational Complexity*, 26(3):717–763, 2017. arXiv:1508.00690.

[JSV04] Mark Jerrum, Alistair Sinclair, and Eric Vigoda. A polynomial-time approximation algorithm for the permanent of a matrix with nonnegative entries. *Journal of the ACM (JACM)*, 51(4):671–697, 2004.

[KK96] Bahman Kalantari and Leonid Khachiyan. On the complexity of nonnegative-matrix scaling. *Linear Algebra and its Applications*, 240:87 – 103, 1996. URL: http://www.sciencedirect.com/science/article/pii/002437959400188X, doi:https://doi.org/10.1016/0024-3795(94)00188-X.

[KLLR17] Tsz Chiu Kwok, Lap Chi Lau, Yin Tat Lee, and Akshay Ramachandran. The Paulsen problem, continuous operator scaling, and smoothed analysis. *arXiv preprint arXiv:1710.02587*, 2017.

[KN79] George Kempf and Linda Ness. The length of vectors in representation spaces. In *Algebraic Geometry, Lecture Notes in Math.*, pages 233–243. 1979.

[Kru37] J Kruithof. Telefoonverkeersrekening. *De Ingenieur*, 52:E15–E25, 1937.

[Lan15] J. M. Landsberg. Geometric complexity theory: an introduction for geometers. *ANNALI DELL'UNIVERSITA' DI FERRARA*, 61(1):65–117, May 2015. URL: https://doi.org/10.1007/s11565-014-0202-7, doi:10.1007/s11565-014-0202-7.

[Lie90] Elliot Lieb. Gaussian kernels have only Gaussian maximizers. *Inventiones Mathematicae*, 102:179–208, 1990.

[LSW98] Nati Linial, Alex Samorodnitsky, and Avi Wigderson. A deterministic strongly polynomial algorithm for matrix scaling and approximate permanents. *STOC*, pages 644–652, 1998.

[MS02] Ketan D. Mulmuley and Milind Sohoni. Geometric complexity theory i: An approach to the p vs. np and related problems. *SIAM J. Comput.*, 31(2):496–526, February 2002. URL: https://doi.org/10.1137/S009753970038715X, doi:10.1137/S009753970038715X.

[Mum65] David Mumford. *Geometric invariant theory*. Springer-Verlag, Berlin-New York, 1965.

[NM84] Linda Ness and David Mumford. A stratification of the null cone via the moment map. *American Journal of Mathematics*, 106(6):1281–1329, 1984. doi:10.2307/2374395.

[RS89] Uriel G. Rothblum and Hans Schneider. Scalings of matrices which have prespecified row sums and column sums via optimization. *Linear Algebra and its Applications*, 114-115:737 – 764, 1989. Special Issue Dedicated to Alan J. Hoffman. URL: http://www.sciencedirect.com/science/article/pii/0024379589904916, doi:https://doi.org/10.1016/0024-3795(89)90491-6.

[SdB01] Aidan Schofield and Michel Van den Bergh. Semi-invariants of quivers for arbitrary dimension vectors. *Indagationes Mathematicae*, 12(1):125–138, 2001.

[Sin64] R. Sinkhorn. A relationship between arbitrary positive matrices and doubly stochastic matrices. *The Annals of Mathematical Statistics*, 35:876–879, 1964.

[Sto62] Richard Stone. Multiple classifications in social accounting. *Bulletin de l'Institut International de Statistique*, 39(3):215–233, 1962.

[Stu08] Bernd Sturmfels. *Algorithms in invariant theory*. Springer, 2008.

[SZ90] Michael H Schneider and Stavros A Zenios. A comparative study of algorithms for matrix balancing. *Operations research*, 38(3):439–455, 1990.

[Tao16] Terence Tao. A symmetric formulation of the Croot-Lev-Pach-Ellenberg-Gijswijt capset bound. Blog post, 2016. https://terrytao.wordpress.com/2016/05/18/a-symmetric-formulation-of-the-croot-lev-pach-ellenberg-gijswijt-capset-bound.

[Val79a] Leslie G Valiant. Completeness classes in algebra. In *Proceedings of the eleventh annual ACM symposium on Theory of computing*, pages 249–261. ACM, 1979.

[Val79b] Leslie G Valiant. The complexity of computing the permanent. *Theoretical computer science*, 8(2):189–201, 1979.

[VDDM03] Frank Verstraete, Jeroen Dehaene, and Bart De Moor. Normal forms and entanglement measures for multipartite quantum states. *Physical Review A*, 68(1):012103, 2003. `arXiv:quant-ph/0105090`, `doi:10.1103/PhysRevA.68.012103`.

[Woo11] Chris Woodward. Moment maps and geometric invariant theory. *Lecture notes, arXiv:0912.1132*, 2011.