(Hidden) Symmetries of Computational Problems

Michael Walter (Ruhr-Uni Bochum)

Michèle in Action, September 2023

joint works with P. Bürgisser, L. Dogan, C. Franks, A. Garg, V. Makam, H. Nieuwboer, R. Oliveira, A. Ramachandran, F. Witteveen, A. Wigderson, . . .

RUHR UNIVERSITÄT BOCHUM RUB

European Research Council
Established by the European Commission

Prelude: Matrix scaling

Let X be matrix with nonnegative entries. A *scaling* of X is a matrix

$$\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix} X \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_n \end{pmatrix} \qquad (a_1, \ldots, b_n > 0).$$

It is called *doubly stochastic* if row & column sums are 1.

Matrix scaling problem: Given X, find approx. doubly stochastic scalings.

Sinkhorn algorithm: Alternatingly normalize rows & columns:

$$\begin{pmatrix} 1 & 2 \\ 4 & 0 \end{pmatrix} \xrightarrow{\text{rows}} \begin{pmatrix} 1/3 & 2/3 \\ 1 & 0 \end{pmatrix} \xrightarrow{\text{cols}} \begin{pmatrix} 1/4 & 1 \\ 3/4 & 0 \end{pmatrix} \longrightarrow \ldots \longrightarrow \begin{pmatrix} \varepsilon & 1 \\ 1-\varepsilon & 0 \end{pmatrix}$$

- This converges whenever possible, and in polynomial time! [LSW]
- Possible iff bipartite perfect matching in support of X.

Applications to statistics, combinatorics, numerics, complexity, machine learning, ...

Prelude: Matrix scaling

Let X be matrix with nonnegative entries. A *scaling* of X is a matrix

$$\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix} X \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_n \end{pmatrix} \qquad (a_1, \dots, b_n > 0).$$

It is called *doubly stochastic* if row & column sums are 1.

Matrix scaling problem: Given X, find approx. doubly stochastic scalings.

Sinkhorn algorithm: Alternatingly normalize rows & columns:

$$\begin{pmatrix} 1 & 2 \\ 4 & 0 \end{pmatrix} \xrightarrow{\text{rows}} \begin{pmatrix} 1/3 & 2/3 \\ 1 & 0 \end{pmatrix} \xrightarrow{\text{cols}} \begin{pmatrix} 1/4 & 1 \\ 3/4 & 0 \end{pmatrix} \longrightarrow \dots \longrightarrow \begin{pmatrix} \varepsilon & 1 \\ 1-\varepsilon & 0 \end{pmatrix}$$

- This converges whenever possible, and in polynomial time! [LSW]
- Possible iff bipartite perfect matching in support of X.

Applications to statistics, combinatorics, numerics, complexity, machine learning, ...

Prelude: Matrix scaling

Let X be matrix with nonnegative entries. A *scaling* of X is a matrix

$$\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix} X \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_n \end{pmatrix} \qquad (a_1, \dots, b_n > 0).$$

It is called *doubly stochastic* if row & column sums are 1.

Matrix scaling problem: Given X, find approx. doubly stochastic scalings.

Sinkhorn algorithm: Alternatingly normalize rows & columns:

$$\begin{pmatrix} 1 & 2 \\ 4 & 0 \end{pmatrix} \xrightarrow{\text{rows}} \begin{pmatrix} 1/3 & 2/3 \\ 1 & 0 \end{pmatrix} \xrightarrow{\text{cols}} \begin{pmatrix} 1/4 & 1 \\ 3/4 & 0 \end{pmatrix} \longrightarrow \dots \longrightarrow \begin{pmatrix} \varepsilon & 1 \\ 1-\varepsilon & 0 \end{pmatrix}$$

- This converges whenever possible, and in polynomial time! [LSW]
- Possible iff bipartite perfect matching in support of X.

Applications to statistics, combinatorics, numerics, complexity, machine learning, ...

Prelude: Matrix scaling

Let X be matrix with nonnegative entries. A *scaling* of X is a matrix

$$\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix} X \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_n \end{pmatrix} \qquad (a_1, \ldots, b_n > 0).$$

It is called *doubly stochastic* if row & column sums are 1.

Matrix scaling problem: Given X, find approx. doubly stochastic scalings.

Sinkhorn algorithm: Alternatingly normalize rows & columns:

$$\begin{pmatrix} 1 & 2 \\ 4 & 0 \end{pmatrix} \xrightarrow{\text{rows}} \begin{pmatrix} 1/3 & 2/3 \\ 1 & 0 \end{pmatrix} \xrightarrow{\text{cols}} \begin{pmatrix} 1/4 & 1 \\ 3/4 & 0 \end{pmatrix} \longrightarrow \ldots \longrightarrow \begin{pmatrix} \varepsilon & 1 \\ 1-\varepsilon & 0 \end{pmatrix}$$

- This converges whenever possible, and in polynomial time! [LSW]
- Possible iff bipartite perfect matching in support of X.

Applications to statistics, combinatorics, numerics, complexity, machine learning, . . .

Prelude: Matrix scaling

Let X be matrix with nonnegative entries. A *scaling* of X is a matrix

$$\begin{pmatrix} a_1 & & \\ & \ddots & \\ & & a_n \end{pmatrix} X \begin{pmatrix} b_1 & & \\ & \ddots & \\ & & b_n \end{pmatrix} \qquad (a_1, \ldots, b_n > 0).$$

It is called *doubly stochastic* if row & column sums are 1.

Matrix scaling problem: Given X, find approx. doubly stochastic scalings.

Sinkhorn algorithm: Alternatingly normalize rows & columns:

- Why does such a simple "greedy" algorithm work?
- What's the deal with perfect matchings?
- Answer to both: **Hidden symmetry** – the permanent of X. *And this is not an accident...*

Applications to statistics, combinatorics, numerics, complexity, machine learning, ...

Overview

There are geometric and algebraic problems, arising from group actions, that are amenable to convex optimization on symmetric spaces.

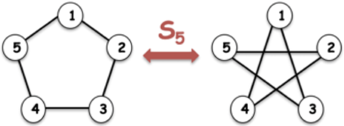
Scaling & marginal problems	$\longleftrightarrow$	Norm minimization

These are connected to a wide range of problems in mathematics and computer science that at first glance might appear unrelated.

Plan for today:

1. Introduction to the setting
2. Applications and connections
3. Algorithmic solutions

Symmetries and group actions



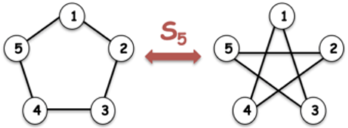
Problem: How can we algorithmically and efficiently check equivalence?

Interesting (and often difficult) problems with many applications:

- no polynomial-time algorithm known for graph isomorphism
- matrices equivalent iff equal rank, but how about tensors?
- given an arithmetic formula, does it always compute zero? finding a deterministic algorithm is famous open problem in computer science
- computing *normal forms*, describing *moduli spaces* and *invariants*...

In many applications, *equivalence* can be modeled by *group action*...

Symmetries and group actions



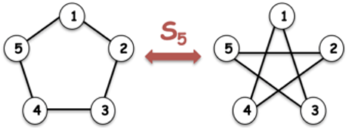
Problem: How can we algorithmically and efficiently check equivalence?

Interesting (and often difficult) problems with many applications:

- no polynomial-time algorithm known for graph isomorphism
- matrices equivalent iff equal rank, but how about tensors?
- given an arithmetic formula, does it always compute zero? finding a deterministic algorithm is famous open problem in computer science
- computing *normal forms*, describing *moduli spaces* and *invariants*...

In many applications, *equivalence* can be modeled by *group action*...

Symmetries and group actions



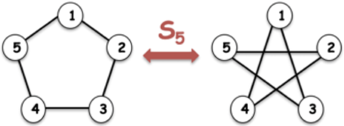
Problem: How can we algorithmically and efficiently check equivalence?

Interesting (and often difficult) problems with many applications:

- no polynomial-time algorithm known for graph isomorphism
- matrices equivalent iff equal rank, but how about tensors?
- given an arithmetic formula, does it always compute zero? finding a deterministic algorithm is famous open problem in computer science
- computing *normal forms*, describing *moduli spaces* and *invariants*. . .

In many applications, *equivalence* can be modeled by *group action*. . .

Symmetries and group actions



Problem: How can we algorithmically and efficiently check equivalence?

Interesting (and often difficult) problems with many applications:

- no polynomial-time algorithm known for graph isomorphism
- matrices equivalent iff equal rank, but how about tensors?
- given an arithmetic formula, does it always compute zero? finding a deterministic algorithm is famous open problem in computer science
- computing *normal forms*, describing *moduli spaces* and *invariants*...

In many applications, *equivalence* can be modeled by *group action*...

Examples: Group actions and symmetries

- Matrices and want to ignore basis: GL(n) by conjugation

$$g \cdot M := gMg^{-1}$$

- Tensor networks: simultaneous conjugation on matrix tuples

$$g \cdot (M_j) := (gM_jg^{-1})$$

- Quantum channels: left-right action on matrix tuples

$$(g,h) \cdot (M_j) := (gM_jh^T) \qquad \Phi(X) = \sum_j M_jXM_j^*$$

- The above are also examples of quiver representations.
- Quantum states, but also bilinear problems: tensor action

$$(A,B,C) \cdot T := (A \otimes B \otimes C)T$$

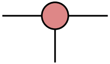
Examples: Group actions and symmetries

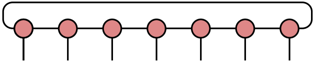
- Matrices and want to ignore basis: $\mathrm{GL}(n)$ by conjugation

$$g \cdot M := gMg^{-1}$$

- Tensor networks: simultaneous conjugation on matrix tuples

$$g \cdot (M_j) := (gM_jg^{-1})$$





- Quantum channels: left-right action on matrix tuples

$$(g, h) \cdot (M_j) := (gM_jh^T) \qquad \Phi(X) = \sum_j M_jXM_j^*$$

- The above are also examples of quiver representations.
- Quantum states, but also bilinear problems: tensor action

$$(A, B, C) \cdot T := (A \otimes B \otimes C)T$$

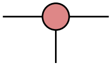
Examples: Group actions and symmetries

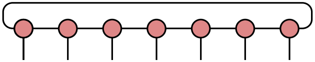
- Matrices and want to ignore basis: $\mathrm{GL}(n)$ by conjugation

$$g \cdot M := gMg^{-1}$$

- Tensor networks: simultaneous conjugation on matrix tuples

$$g \cdot (M_j) := (gM_jg^{-1})$$





- Quantum channels: left-right action on matrix tuples

$$(g,h) \cdot (M_j) := (gM_jh^T) \qquad \Phi(X) = \sum_j M_jXM_j^*$$

- The above are also examples of quiver representations.
- Quantum states, but also bilinear problems: tensor action

$$(A,B,C) \cdot T := (A \otimes B \otimes C)T$$

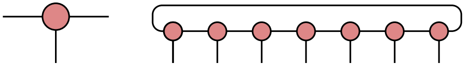
Examples: Group actions and symmetries

- Matrices and want to ignore basis: $\mathrm{GL}(n)$ by conjugation

$$g \cdot M := gMg^{-1}$$

- Tensor networks: simultaneous conjugation on matrix tuples

$$g \cdot (M_j) := (gM_jg^{-1})$$



- Quantum channels: left-right action on matrix tuples

$$(g, h) \cdot (M_j) := (gM_jh^T) \qquad \Phi(X) = \sum_j M_j X M_j^*$$

- The above are also examples of quiver representations.
- Quantum states, but also bilinear problems: tensor action

$$(A, B, C) \cdot T := (A \otimes B \otimes C)T$$

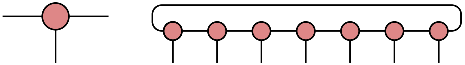
Examples: Group actions and symmetries

- Matrices and want to ignore basis: $\mathrm{GL}(n)$ by conjugation

$$g \cdot M := gMg^{-1}$$

- Tensor networks: simultaneous conjugation on matrix tuples

$$g \cdot (M_j) := (gM_jg^{-1})$$

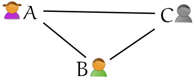


- Quantum channels: left-right action on matrix tuples

$$(g, h) \cdot (M_j) := (gM_jh^T) \qquad \Phi(X) = \sum_j M_j X M_j^*$$

- The above are also examples of quiver representations.
- Quantum states, but also bilinear problems: tensor action

$$(A, B, C) \cdot T := (A \otimes B \otimes C)T$$

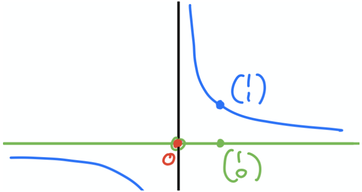


Formalizing equivalence: Orbit problems

Group $G \subseteq \mathrm{GL}_n(\mathbb{C})$ complex reductive, such as GL_n, SL_n, or $\mathrm{T}_n = \left(\begin{smallmatrix} \cdot & \\ & \cdot \end{smallmatrix}\right)$

Action on $V = \mathbb{C}^m$ by linear transformations

Orbits $Gv = \{g \cdot v \ : \ g \in G\}$ and their closures $\overline{Gv}$



Orbit problems:

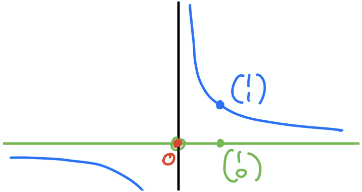
- Given v and w, are they in the same orbit? That is, is $Gv = Gw$?
- Robust versions: Is $w \in \overline{Gv}$? Is $\overline{Gv} \cap \overline{Gw} \neq \emptyset$?
- **Null cone problem:** $0 \in \overline{Gv}$?

Formalizing equivalence: Orbit problems

Group $G \subseteq \mathrm{GL}_n(\mathbb{C})$ complex reductive, such as GL_n, SL_n, or $\mathrm{T}_n = \left(\ddots\right)$

Action on $V = \mathbb{C}^m$ by linear transformations

Orbits $Gv = \{g \cdot v \; : \; g \in G\}$ and their closures $\overline{Gv}$



Orbit problems:

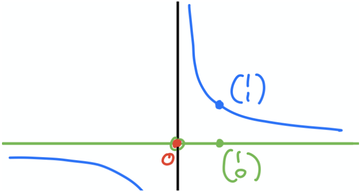
- Given v and w, are they in the same orbit? That is, is $Gv = Gw$?
- Robust versions: Is $w \in \overline{Gv}$? Is $\overline{Gv} \cap \overline{Gw} \neq \emptyset$?
- **Null cone problem:** $0 \in \overline{Gv}$?

Formalizing equivalence: Orbit problems

Group $G \subseteq \mathrm{GL}_n(\mathbb{C})$ complex reductive, such as GL_n, SL_n, or $\mathrm{T}_n = \left(\ddots\right)$

Action on $V = \mathbb{C}^m$ by linear transformations

Orbits $Gv = \{g \cdot v \,:\, g \in G\}$ and their closures $\overline{Gv}$



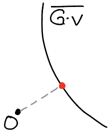
Orbit problems:

- Given v and w, are they in the same orbit? That is, is $Gv = Gw$?
- Robust versions: Is $w \in \overline{Gv}$? Is $\overline{Gv} \cap \overline{Gw} \neq \emptyset$?
- **Null cone problem:** $0 \in \overline{Gv}$?

Big picture: Orbits, optimization, and scaling

For concreteness, focus on **null cone problem**:

Is $0 \in \overline{Gv}$?

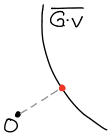


Big picture: Orbits, optimization, and scaling

Is $P(v) = P(0)$ for every *invariant* polynomial P? Algebra

$\Updownarrow$

Is $0 \in \overline{Gv}$?



Big picture: Orbits, optimization, and scaling

Is $P(v) = P(0)$ for every *invariant* polynomial P?

Algebra

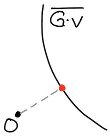
$\Updownarrow$

Is $0 \in \overline{Gv}$?

$\Uparrow$

Minimize $\|g \cdot v\|$ over $g \in G$.

Optimization

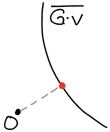


Big picture: Orbits, optimization, and scaling

Is $P(v) = P(0)$ for every *invariant* polynomial P? — Algebra

$\Updownarrow$

Is $0 \in \overline{Gv}$?



$\Nearrow$

Minimize $\|g \cdot v\|$ over $g \in G$. $\Longrightarrow$ Find $g \in G$ s.th. $\nabla_g \|g \cdot v\| \approx 0$.

Optimization

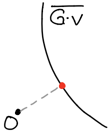
"Scaling Problem"

Big picture: Orbits, optimization, and scaling

Is $P(v) = P(0)$ for every *invariant* polynomial P? — Algebra

$\Updownarrow$

Is $0 \in \overline{Gv}$?



$\Uparrow$

Minimize $\|g \cdot v\|$ over $g \in G$. — Optimization

$\overset{\text{Kempf-Ness}}{\underset{\text{convexity}}{\Longleftrightarrow}}$

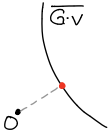
Find $g \in G$ s.th. $\nabla_g \|g \cdot v\| \approx 0$. — "Scaling Problem"

Big picture: Orbits, optimization, and scaling

Is $P(v) = P(0)$ for every *invariant* polynomial P? — Algebra

$\Updownarrow$

Is $0 \in \overline{Gv}$?



Minimize $\|g \cdot v\|$ over $g \in G$. $\overset{\text{Kempf-Ness}}{\underset{\text{convexity}}{\Longleftrightarrow}}$ Find $g \in G$ s.th. $\nabla_g \|g \cdot v\| \approx 0$.

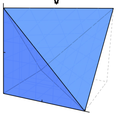
Optimization (which $\Rightarrow$ Is $0 \in \overline{Gv}$?)

$\Downarrow$

Natural normal forms to tackle other orbit problems.

"Scaling Problem"

symplectic $\Updownarrow$ geometry



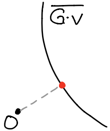
Polytopes

Big picture: Orbits, optimization, and scaling

Is $P(v) = P(0)$ for every *invariant* polynomial P? — Algebra

$\Updownarrow$

Is $0 \in \overline{Gv}$?



$\Uparrow$

Minimize $\|g \cdot v\|$ over $g \in G$. $\xLeftrightarrow[\text{convexity}]{\text{Kempf-Ness}}$ Find $g \in G$ s.th. $\nabla_g \|g \cdot v\| \approx 0$.

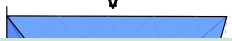
Optimization — "Scaling Problem"

$\Downarrow$ — symplectic $\Updownarrow$ geometry

Na…

oth…

Why care? Surprisingly many problems fit this picture. Make it *quantitative* to obtain new insights and efficient algorithms.



Polytopes

Example: Matrix scaling revisited

Let $G = \mathrm{T}_n \times \mathrm{T}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g,h) \cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g,h) \cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y \in \mathbb{R}^n} \sum_{i,j} |M_{ij}|^2 e^{x_i + y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big)$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

Matrix scaling and norm minimization are equivalent by convexity! ☺
Explains Sinkhorn & permanent. Starting point for cutting-edge algos.

Example: Matrix scaling revisited

Let $G = \mathrm{T}_n \times \mathrm{T}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g,h)\cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g,h)\cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y\in\mathbb{R}^n} \sum_{i,j} |M_{ij}|^2 e^{x_i+y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big)$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

Matrix scaling and norm minimization are equivalent by convexity! ☺
Explains Sinkhorn & permanent. Starting point for cutting-edge algos.

Example: Matrix scaling revisited

Let $G = \mathrm{T}_n \times \mathrm{T}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g, h) \cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g, h) \cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y \in \mathbb{R}^n} \sum_{i,j} |M_{ij}|^2 e^{x_i + y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big)$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

Matrix scaling and norm minimization are equivalent by convexity! ☺
Explains Sinkhorn & permanent. Starting point for cutting-edge algos.

Example: Matrix scaling revisited

Let $G = \mathrm{T}_n \times \mathrm{T}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g,h)\cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g,h)\cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y\in\mathbb{R}^n} \sum_{i,j} |M_{ij}|^2 e^{x_i+y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big)$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

Matrix scaling and norm minimization are equivalent by convexity! ☺
Explains Sinkhorn & permanent. Starting point for cutting-edge algos.

Example: Matrix scaling revisited

Let $G = \mathrm{ST}_n \times \mathrm{ST}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g,h) \cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g,h) \cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y \in \mathbb{R}^n_{\sum=0}} \sum_{i,j} |M_{ij}|^2 e^{x_i + y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big) - \|M\|^2 (\mathbf{1}, \mathbf{1})$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

Matrix scaling and norm minimization are equivalent by convexity! ☺
Explains Sinkhorn & permanent. Starting point for cutting-edge algos.

Example: Matrix scaling revisited

Let $G = \mathrm{ST}_n \times \mathrm{ST}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g,h) \cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g,h) \cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y \in \mathbb{R}^n_{\sum=0}} \sum_{i,j} |M_{ij}|^2 e^{x_i + y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big) - \|M\|^2 (\mathbf{1}, \mathbf{1})$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

Matrix scaling and norm minimization are equivalent by convexity! ☺
Explains Sinkhorn & permanent. Starting point for cutting-edge algos.

Example: Matrix scaling revisited

Let $G = \mathrm{ST}_n \times \mathrm{ST}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g,h)\cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g,h)\cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y\in\mathbb{R}^n_{\sum=0}} \sum_{i,j} |M_{ij}|^2 e^{x_i+y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big) - \|M\|^2 (\mathbf{1}, \mathbf{1})$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

GIT vs symplectic quotient: Given semistable M, Sinkhorn algorithm approximates $M' \in \overline{G\cdot M}$ with zero moment map...

Example: Matrix scaling revisited

Let $G = \mathrm{ST}_n \times \mathrm{ST}_n$ act on $V = \mathrm{Mat}_n(\mathbb{C})$:

$$(g,h)\cdot M = \begin{pmatrix} g_1 & & \\ & \ddots & \\ & & g_n \end{pmatrix} M \begin{pmatrix} h_1 & & \\ & \ddots & \\ & & h_n \end{pmatrix}$$

Norm minimization:

$$\inf_{g,h} \|(g,h)\cdot M\|^2 = \inf_{g,h} \sum_{i,j} |g_i M_{ij} h_j|^2 = \inf_{x,y \in \mathbb{R}^n_{\sum=0}} \sum_{i,j} |M_{ij}|^2 e^{x_i + y_j}$$

- log-convex in x and y

Gradient:

$$\nabla_{x=y=0}(\ldots) = \big(\mathbf{r}(M), \mathbf{c}(M)\big) - \|M\|^2 (\mathbf{1}, \mathbf{1})$$

where $\mathbf{r}(M)$, $\mathbf{c}(M)$ row and column sums of matrix with entries $|M_{ij}|^2$.

In general, torus actions capture linear programming – one of the most widely used paradigms in convex optimization (and more)!

Example: Horn problem

Let $\alpha_1 \geqslant \ldots \geqslant \alpha_n$, $\beta_1 \geqslant \ldots \geqslant \beta_n$, $\gamma_1 \geqslant \ldots \geqslant \gamma_n$ be integers.

> Horn problem: When $\exists$ Hermitian $n \times n$ matrices A, B, C with spectrum α, β, γ such that $A + B = C$?

- Horn conjectured exponentially many linear inequalities on α, β, γ
- e.g., $\alpha_1 + \beta_1 \geqslant \gamma_1$
- proved in late 90s. by now, very precise mathematical understanding.
  [Klyachko, Knutson-Tao, Belkale, Ressayre, Paradan, . . .]

Knutson-Tao: . . . iff *Littlewood-Richardson coefficient* $c^{\gamma}_{\alpha,\beta} > 0$

- counts multiplicities in representation theory, combinatorial gadgets, integer points in polytopes, . . .
- poly-time algorithm can check if this is the case [Mulmuley]

> Yet, no known efficient algorithms to **find** such A, B, C (if α, β, γ are compatible), or to find a violated inequality (if they are not).

Example: Horn problem

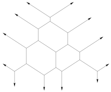
Let $\alpha_1 \geqslant \ldots \geqslant \alpha_n$, $\beta_1 \geqslant \ldots \geqslant \beta_n$, $\gamma_1 \geqslant \ldots \geqslant \gamma_n$ be integers.

> Horn problem: When $\exists$ Hermitian $n \times n$ matrices A, B, C with spectrum α, β, γ such that $A + B = C$?

- Horn conjectured exponentially many linear inequalities on α, β, γ
- e.g., $\alpha_1 + \beta_1 \geqslant \gamma_1$
- proved in late 90s. by now, very precise mathematical understanding.
  [Klyachko, Knutson-Tao, Belkale, Ressayre, Paradan, . . .]

Knutson-Tao: . . . iff *Littlewood-Richardson coefficient* $c_{\alpha,\beta}^{\gamma} > 0$

- counts multiplicities in representation theory, combinatorial gadgets, integer points in polytopes, . . .
- poly-time algorithm can check if this is the case



[Mulmuley]

Yet, no known efficient algorithms to **find** such A, B, C (if α, β, γ are compatible), or to find a violated inequality (if they are not).

Example: Algorithmic Horn problem

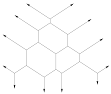
Let $\alpha_1 \geqslant \ldots \geqslant \alpha_n$, $\beta_1 \geqslant \ldots \geqslant \beta_n$, $\gamma_1 \geqslant \ldots \geqslant \gamma_n$ be integers.

> Horn problem: When $\exists$ Hermitian $n \times n$ matrices A, B, C with spectrum α, β, γ such that $A + B = C$?

- Horn conjectured exponentially many linear inequalities on α, β, γ
- e.g., $\alpha_1 + \beta_1 \geqslant \gamma_1$
- proved in late 90s. by now, very precise mathematical understanding.
  [Klyachko, Knutson-Tao, Belkale, Ressayre, Paradan, ...]

Knutson-Tao: ... iff *Littlewood-Richardson coefficient* $c^{\gamma}_{\alpha,\beta} > 0$

- counts multiplicities in representation theory, combinatorial gadgets, integer points in polytopes, ...
- poly-time algorithm can check if this is the case



[Mulmuley]

> Yet, no known efficient algorithms to **find** such A, B, C (if α, β, γ are compatible), or to find a violated inequality (if they are not).

Example: Operator scaling and polynomial identity testing

Any matrix tuple $X = (X_k)$ defines map $\Phi(A) = \sum_k X_k A X_k^*$. *Scaling:*

$$(g, h) \cdot (X_k) = (g X_k h^T) \qquad (g, h \in \mathrm{GL}_n)$$

Doubly stochastic if $\Phi(I) = I$ and $\Phi^*(I) = I$.

Operator scaling problem: Given X, find approx. doubly stochastic scaling.

Possible iff $\det \sum_k \alpha_k \otimes X_k \neq 0$ for some matrices α_k (semi-invariants!).

- equivalent: is arithmetic formula in *non-commuting variables* nonzero?
- a simple algo solves this in deterministic poly-time! [Garg et al, Ivanyos et al]

When α_k restricted to scalars: major open problem in computer science!

Many further connections (Brascamp-Lieb inequalities, Paulsen problem, MLE, . . .).

Example: Operator scaling and polynomial identity testing

Any matrix tuple $X = (X_k)$ defines map $\Phi(A) = \sum_k X_k A X_k^*$. *Scaling:*

$$(g, h) \cdot (X_k) = (g X_k h^T) \qquad (g, h \in \mathrm{GL}_n)$$

Doubly stochastic if $\Phi(I) = I$ and $\Phi^*(I) = I$.

Operator scaling problem: Given X, find approx. doubly stochastic scaling.

Possible iff $\det \sum_k \alpha_k \otimes X_k \neq 0$ for some matrices α_k (semi-invariants!).

- equivalent: is arithmetic formula in *non-commuting variables* nonzero?
- a simple algo solves this in deterministic poly-time! [Garg et al, Ivanyos et al]

When α_k restricted to scalars: major open problem in computer science!

Many further connections (Brascamp-Lieb inequalities, Paulsen problem, MLE, . . .).

Example: Operator scaling and polynomial identity testing

Any matrix tuple $X = (X_k)$ defines map $\Phi(A) = \sum_k X_k A X_k^*$. *Scaling:*

$$(g, h) \cdot (X_k) = (g X_k h^T) \qquad (g, h \in \mathrm{GL}_n)$$

Doubly stochastic if $\Phi(I) = I$ and $\Phi^*(I) = I$.

Operator scaling problem: Given X, find approx. doubly stochastic scaling.

Possible iff $\det \sum_k \alpha_k \otimes X_k \neq 0$ for some matrices α_k (semi-invariants!).

- equivalent: is arithmetic formula in *non-commuting variables* nonzero?
- a simple algo solves this in deterministic poly-time! [Garg et al, Ivanyos et al]

When α_k restricted to scalars: major open problem in computer science!

Many further connections (Brascamp-Lieb inequalities, Paulsen problem, MLE, ...).

Example: Operator scaling and polynomial identity testing

Any matrix tuple $X = (X_k)$ defines map $\Phi(A) = \sum_k X_k A X_k^*$. *Scaling:*

$$(g, h) \cdot (X_k) = (g X_k h^T) \qquad (g, h \in \mathrm{GL}_n)$$

Doubly stochastic if $\Phi(I) = I$ and $\Phi^*(I) = I$.

Operator scaling problem: Given X, find approx. doubly stochastic scaling.

Possible iff $\det \sum_k \alpha_k \otimes X_k \neq 0$ for some matrices α_k (semi-invariants!).

- equivalent: is arithmetic formula in *non-commuting variables* nonzero?
- a simple algo solves this in deterministic poly-time! [Garg et al, Ivanyos et al]

When α_k restricted to scalars: major open problem in computer science!

Many further connections (Brascamp-Lieb inequalities, Paulsen problem, MLE, . . .).

Example: Operator scaling and polynomial identity testing

Any matrix tuple $X = (X_k)$ defines map $\Phi(A) = \sum_k X_k A X_k^*$. *Scaling:*

$$(g, h) \cdot (X_k) = (g X_k h^T) \qquad (g, h \in \mathrm{GL}_n)$$

Doubly stochastic if $\Phi(I) = I$ and $\Phi^*(I) = I$.

Operator scaling problem: Given X, find approx. doubly stochastic scaling.

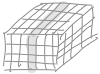
Possible iff $\det \sum_k \alpha_k \otimes X_k \neq 0$ for some matrices α_k (semi-invariants!).

- equivalent: is arithmetic formula in *non-commuting variables* nonzero?
- a simple algo solves this in deterministic poly-time! [Garg et al, Ivanyos et al]

When α_k restricted to scalars: major open problem in computer science!

Many further connections (Brascamp-Lieb inequalities, Paulsen problem, MLE, . . .).

Example: Tensor scaling



Tensors $T \in (\mathbb{C}^n)^{\otimes d}$ describe quantum states of d particles. *Scaling:*

$$(g_1, \ldots, g_d) \cdot T := (g_1 \otimes \cdots \otimes g_d) T \qquad (g_k \in \mathrm{GL}_n)$$

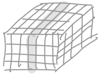
- state of k-th particle is $\sigma_k = S_k S_k^*$, where S_k is k-th flattening of S.

Tensor scaling problem: Given T, which $(\sigma_1, \ldots, \sigma_d)$ can be obtained by scaling?

- since $[S] \mapsto (\sigma_1, \ldots, \sigma_d)$ is a moment map for action of $K = U(n)^d$, answer captured by moment polytopes
- distance to origin measures instability → recent notions such as *quantum functionals*, *G-stable ranks*. [Blasiak et al, Christandl et al, Derksen, ...]
- applications in algebraic complexity, combinatorics, q. information, ...
- exponentially complex polytopes, succinctly encoded by group action

What other interesting polytopes are hidden in this way?

Example: Tensor scaling



Tensors $T \in (\mathbb{C}^n)^{\otimes d}$ describe quantum states of d particles. *Scaling:*

$$S = (g_1, \ldots, g_d) \cdot T := (g_1 \otimes \cdots \otimes g_d)\,T \qquad (g_k \in \mathrm{GL}_n)$$

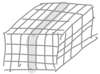
- state of k-th particle is $\sigma_k = S_k S_k^*$, where S_k is k-th flattening of S.

Tensor scaling problem: Given T, which $(\sigma_1, \ldots, \sigma_d)$ can be obtained by scaling?

- since $[S] \mapsto (\sigma_1, \ldots, \sigma_d)$ is a moment map for action of $K = U(n)^d$, answer captured by moment polytopes
- distance to origin measures instability $\to$ recent notions such as *quantum functionals*, *G-stable ranks*. [Blasiak et al, Christandl et al, Derksen, ...]
- applications in algebraic complexity, combinatorics, q. information, ...
- exponentially complex polytopes, succinctly encoded by group action

What other interesting polytopes are hidden in this way?

Example: Tensor scaling



Tensors $T \in (\mathbb{C}^n)^{\otimes d}$ describe quantum states of d particles. *Scaling:*

$$S = (g_1, \ldots, g_d) \cdot T := (g_1 \otimes \cdots \otimes g_d)\, T \qquad (g_k \in \mathrm{GL}_n)$$

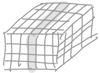
- state of k-th particle is $\sigma_k = S_k S_k^*$, where S_k is k-th flattening of S.

Tensor scaling problem: Given T, which $(\sigma_1, \ldots, \sigma_d)$ can be obtained by scaling?

- since $[S] \mapsto (\sigma_1, \ldots, \sigma_d)$ is a moment map for action of $K = U(n)^d$, answer captured by moment polytopes
- distance to origin measures instability $\rightarrow$ recent notions such as *quantum functionals*, *G-stable ranks*. [Blasiak et al, Christandl et al, Derksen, ...]
- applications in algebraic complexity, combinatorics, q. information, ...
- exponentially complex polytopes, succinctly encoded by group action

What other interesting polytopes are hidden in this way?

Example: Tensor scaling

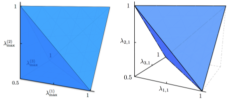


Tensors $T \in (\mathbb{C}^n)^{\otimes d}$ describe quantum states of d particles. *Scaling:*

$$S = (g_1, \ldots, g_d) \cdot T := (g_1 \otimes \cdots \otimes g_d)\, T \qquad (g_k \in \mathrm{GL}_n)$$

- state of k-th particle is $\sigma_k = S_k S_k^*$, where S_k is k-th flattening of S.

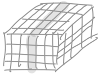
Tensor scaling problem: Given T, which $(\sigma_1, \ldots, \sigma_d)$ can be obtained by scaling?



- since $[S] \mapsto (\sigma_1, \ldots, \sigma_d)$ is a moment map for action of $K = U(n)^d$, answer captured by moment polytopes
- distance to origin measures instability $\to$ recent notions such as *quantum functionals*, *G-stable ranks*. [Blasiak et al, Christandl et al, Derksen, ...]
- applications in algebraic complexity, combinatorics, q. information, ...
- exponentially complex polytopes, succinctly encoded by group action

What other interesting polytopes are hidden in this way?

Example: Tensor scaling

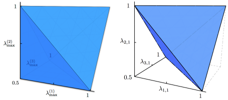


Tensors $T \in (\mathbb{C}^n)^{\otimes d}$ describe quantum states of d particles. *Scaling:*

$$S = (g_1, \ldots, g_d) \cdot T := (g_1 \otimes \cdots \otimes g_d)\, T \qquad (g_k \in \mathrm{GL}_n)$$

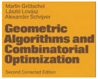
- state of k-th particle is $\sigma_k = S_k S_k^*$, where S_k is k-th flattening of S.

Tensor scaling problem: Given T, which $(\sigma_1, \ldots, \sigma_d)$ can be obtained by scaling?

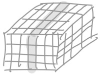


- since $[S] \mapsto (\sigma_1, \ldots, \sigma_d)$ is a moment map for action of $K = U(n)^d$, answer captured by moment polytopes
- distance to origin measures instability $\rightarrow$ recent notions such as *quantum functionals*, *G-stable ranks*. [Blasiak et al, Christandl et al, Derksen, ...]
- applications in algebraic complexity, combinatorics, q. information, ...
- exponentially complex polytopes, succinctly encoded by group action

What other interesting polytopes are hidden in this way?



Example: Tensor scaling

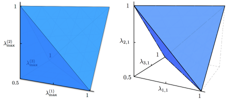


Tensors $T \in (\mathbb{C}^n)^{\otimes d}$ describe quantum states of d particles. *Scaling:*

$$S = (g_1, \ldots, g_d) \cdot T := (g_1 \otimes \cdots \otimes g_d)\, T \qquad (g_k \in \mathrm{GL}_n)$$

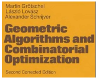
- state of k-th particle is $\sigma_k = S_k S_k^*$, where S_k is k-th flattening of S.

Tensor scaling problem: Given T, which $(\sigma_1, \ldots, \sigma_d)$ can be obtained by scaling?



- since $[S] \mapsto (\sigma_1, \ldots, \sigma_d)$ is a moment map for action of $K = U(n)^d$, answer captured by moment polytopes
- distance to origin measures instability $\to$ recent notions such as *quantum functionals*, *G-stable ranks*. [Blasiak et al, Christandl et al, Derksen, ...]
- applications in algebraic complexity, combinatorics, q. information, ...
- exponentially complex polytopes, succinctly encoded by group action

What other interesting polytopes are hidden in this way?



Connections and applications

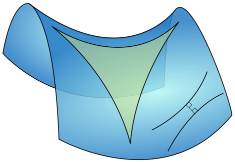
Scaling problems capture and connect many other applications (thanks to often hidden symmetries):

- theoretical: program testing, VP vs VNP
- algebraic: representation theory, moduli spaces
- quantum: quantum entanglement, tensor networks
- classical: derandomization
- continuous: optimal transport
- applied: maximum likelihood
- · · ·

This was discovered in a series of works, by many authors, over the past years. It has already proved key to faster algorithms and structural insight.

> General picture? How to find algorithms beyond Sinkhorn & friends?

Symmetry and Optimization



Norm minimization and gradient

In general, we want to minimize the Kempf-Ness function:

$$F\colon G \to \mathbb{R}, \quad F(g) := \log \|g \cdot v\|$$

For concreteness, $G = \mathrm{GL}_n$. By the polar decomposition, can restrict to:

$$\mathrm{PD}_n = \{p = e^X \;:\; X \in \mathrm{Herm}_n\} \cong K \backslash G$$

This is a Hadamard symmetric space. It has nonpositive curvature:

Gradient $\nabla F(I)$ is *moment map* in the sense of symplectic geometry.
As discussed $\nabla F = 0$ captures natural "scaling" problems!

Norm minimization and gradient

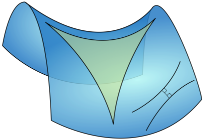
In general, we want to minimize the Kempf-Ness function:

$$F\colon G \to \mathbb{R}, \quad F(g) := \log \|g \cdot v\|$$

For concreteness, $G = \mathrm{GL}_n$. By the polar decomposition, can restrict to:

$$\mathrm{PD}_n = \{p = e^X \ : \ X \in \mathrm{Herm}_n\} \cong K \backslash G$$

This is a Hadamard symmetric space. It has nonpositive curvature:



Gradient $\nabla F(I)$ is *moment map* in the sense of symplectic geometry.
As discussed $\nabla F = 0$ captures natural "scaling" problems!

Norm minimization and gradient

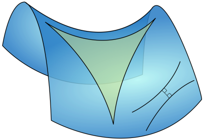
In general, we want to minimize the Kempf-Ness function:

$$F\colon G \to \mathbb{R}, \quad F(g) := \log \|g \cdot v\|$$

For concreteness, $G = \mathrm{GL}_n$. By the polar decomposition, can restrict to:

$$\mathrm{PD}_n = \{p = e^X \;:\; X \in \mathrm{Herm}_n\} \cong K \backslash G$$

This is a Hadamard symmetric space. It has nonpositive curvature:



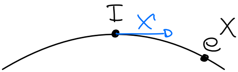
Gradient $\nabla F(I)$ is *moment map* in the sense of symplectic geometry.
As discussed $\nabla F = 0$ captures natural "scaling" problems!

Geodesic convexity

While not convex in the usual sense, the objective

$$F(p) = \log \|p \cdot v\|$$

is convex along the geodesics of PD_n, i.e., $\partial_t^2 F(e^{Xt}) \geq 0$. [Kempf-Ness]



This implies critical points are global minima (like in the Euclidean case).

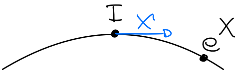
How convex for given action? Necessary for algorithms!

Geodesic convexity

While not convex in the usual sense, the objective

$$F(p) = \log \|p \cdot v\|$$

is convex along the geodesics of PD_n, i.e., $\partial_t^2 F(e^{Xt}) \geqslant 0$. [Kempf-Ness]



This implies critical points are global minima (like in the Euclidean case).

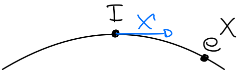
How convex for given action? Necessary for algorithms!

Geodesic convexity made quantitative

While not convex in the usual sense, the objective

$$F(p) = \log \|p \cdot v\|$$

is convex along the geodesics of PD_n, i.e., $\partial_t^2 F(e^{Xt}) \geqslant 0$. [Kempf-Ness]



This implies critical points are global minima (like in the Euclidean case).

How convex for given action? Necessary for algorithms!

Geodesic convexity made quantitative

Easy: The objective $F(p) = \log \|p \cdot v\|$ is "smooth", meaning

$$\partial_t^2 F(e^{Xt}) \leqslant L\|X\|_2^2.$$

Not so easy: Quantitative Kempf-Ness theorem. For $F_* = \inf_p F(p)$,

$$1 - \frac{\|\nabla F\|_2}{\gamma} \leqslant e^{F_* - F} \leqslant 1 - \frac{\|\nabla F\|_2^2}{2L}$$

- ☺ relates norm minimization ⇔ scaling in a quantitative way
- ☺ implies either can solve null cone problem, rigorously!
- ▶ Non-commutative variant of convex duality.

Parameters L, γ depend on representation-theoretic data of action.

Geodesic convexity made quantitative

Easy: The objective $F(p) = \log \|p \cdot v\|$ is "smooth", meaning

$$\partial_t^2 F(e^{Xt}) \leqslant L\|X\|_2^2.$$

Not so easy: Quantitative Kempf-Ness theorem. For $F_* = \inf_p F(p)$,

$$1 - \frac{\|\nabla F\|_2}{\gamma} \leqslant e^{F_* - F} \leqslant 1 - \frac{\|\nabla F\|_2^2}{2L}$$

- ☺ relates norm minimization ⇔ scaling in a quantitative way
- ☺ implies either can solve null cone problem, rigorously!
- ▶ Non-commutative variant of convex duality.

Parameters L, γ depend on representation-theoretic data of action.

Geodesic convexity made quantitative

Easy: The objective $F(p) = \log \|p \cdot v\|$ is "smooth", meaning

$$\partial_t^2 F(e^{Xt}) \leqslant L\|X\|_2^2.$$

Not so easy: Quantitative Kempf-Ness theorem. For $F_* = \inf_p F(p)$,

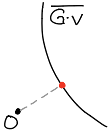
$$1 - \frac{\|\nabla F\|_2}{\gamma} \leqslant e^{F_* - F} \leqslant 1 - \frac{\|\nabla F\|_2^2}{2L}$$

- ☺ relates norm minimization ⇔ scaling in a quantitative way
- ☺ implies either can solve null cone problem, rigorously!
- ► Non-commutative variant of convex duality.

Parameters L, γ depend on representation-theoretic data of action.

Framework: Noncommutative group optimization [BFGOWW]

Action of complex reductive $G \subseteq \mathrm{GL}_n$ on $V \cong \mathbb{C}^m$.



Is $0 \in \overline{Gv}$?

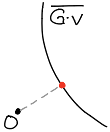
Minimize $\|g \cdot v\|$ over $g \in G$. $\overset{\text{convexity}}{\underset{\text{quantitative}}{\Longleftrightarrow}}$ Find $g \in G$ s.th. $\nabla \|g \cdot v\| \approx 0$.

Norm Minimization — Scaling Problem

- All examples mentioned earlier fall into this framework.
- Geodesic convexity explains why simple greedy algorithms can work.
- Made quantitative by NC generalization of convex programming duality.
- We also provide general algorithms for geodesic convex optimization (which solve problems in poly time for several interesting actions).

Framework: Noncommutative group optimization [BFGOWW]

Action of complex reductive $G \subseteq \mathrm{GL}_n$ on $V \cong \mathbb{C}^m$.



Is $0 \in \overline{Gv}$?

Minimize $\|g \cdot v\|$ over $g \in G$. $\xLeftrightarrow[\text{quantitative}]{\text{convexity}}$ Find $g \in G$ s.th. $\nabla\|g \cdot v\| \approx 0$.

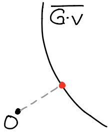
Norm Minimization — Scaling Problem

- All examples mentioned earlier fall into this framework.
- Geodesic convexity explains why simple greedy algorithms can work.
- Made quantitative by NC generalization of convex programming duality.
- We also provide general algorithms for geodesic convex optimization (which solve problems in poly time for several interesting actions).

Framework: Noncommutative group optimization [BFGOWW]

Action of complex reductive $G \subseteq \mathrm{GL}_n$ on $V \cong \mathbb{C}^m$.

Is $0 \in \overline{Gv}$?



Minimize $\|g \cdot v\|$ over $g \in G$. ⟸ convexity / quantitative ⟹ Find $g \in G$ s.th. $\nabla \|g \cdot v\| \approx 0$.

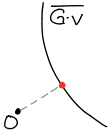
Norm Minimization — Scaling Problem

- All examples mentioned earlier fall into this framework.
- Geodesic convexity explains why simple greedy algorithms can work.
- Made quantitative by NC generalization of convex programming duality.
- We also provide general algorithms for geodesic convex optimization (which solve problems in poly time for several interesting actions).

Framework: Noncommutative group optimization [BFGOWW]

Action of complex reductive $G \subseteq \mathrm{GL}_n$ on $V \cong \mathbb{C}^m$.

Is $0 \in \overline{Gv}$?



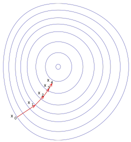
Minimize $\|g \cdot v\|$ over $g \in G$. $\underset{\text{quantitative}}{\overset{\text{convexity}}{\Longleftrightarrow}}$ Find $g \in G$ s.th. $\nabla \|g \cdot v\| \approx 0$.

Norm Minimization — Scaling Problem

- All examples mentioned earlier fall into this framework.
- Geodesic convexity explains why simple greedy algorithms can work.
- Made quantitative by NC generalization of convex programming duality.
- We also provide general algorithms for geodesic convex optimization (which solve problems in poly time for several interesting actions).

Algorithmic appetizer

Simplest approach: Repeatedly perform geodesic gradient steps



$$g \leftarrow e^{-\frac{1}{L}\nabla F(g)} g.$$

Theorem

If F bounded from below, algorithm finds $g \in G$ such that $\|\nabla F(g)\| \leqslant \varepsilon$ within $T = \text{poly}(\frac{1}{\varepsilon}, \text{input size})$ steps.

Analysis: Smoothness implies F decreases in each step. Combine with a priori lower bound obtained using constructive invariant theory.

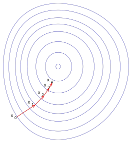
Corollary

Same algorithm solves null cone problem in time $\text{poly}(\frac{1}{\gamma}, \text{input size})$.

Much faster (in theory and practice) than algebraic methods (e.g., via Gröbner bases).

Algorithmic appetizer

Simplest approach: Repeatedly perform geodesic gradient steps



$$g \leftarrow e^{-\frac{1}{L}\nabla F(g)} g.$$

Theorem

If F bounded from below, algorithm finds $g \in G$ such that $\|\nabla F(g)\| \leqslant \varepsilon$ within $T = \text{poly}(\frac{1}{\varepsilon}, \text{input size})$ steps.

Analysis: Smoothness implies F decreases in each step. Combine with a priori lower bound obtained using constructive invariant theory.

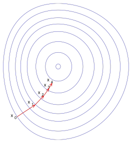
Corollary

Same algorithm solves null cone problem in time $\text{poly}(\frac{1}{\gamma}, \text{input size})$.

Much faster (in theory and practice) than algebraic methods (e.g., via Gröbner bases).

Algorithmic appetizer

Simplest approach: Repeatedly perform geodesic gradient steps



$$g \leftarrow e^{-\frac{1}{L}\nabla F(g)} g.$$

Theorem

If F bounded from below, algorithm finds $g \in G$ such that $\|\nabla F(g)\| \leqslant \varepsilon$ within $T = \text{poly}(\frac{1}{\varepsilon}, \text{input size})$ steps.

Analysis: Smoothness implies F decreases in each step. Combine with a priori lower bound obtained using constructive invariant theory.

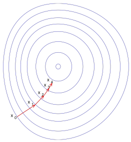
Corollary

Same algorithm solves null cone problem in time $\text{poly}(\frac{1}{\gamma}, \text{input size})$.

Much faster (in theory and practice) than algebraic methods (e.g., via Gröbner bases).

Algorithmic appetizer

Simplest approach: Repeatedly perform geodesic gradient steps



$$g \leftarrow e^{-\frac{1}{L}\nabla F(g)} g.$$

Theorem

If F bounded from below, algorithm finds $g \in G$ such that $\|\nabla F(g)\| \leqslant \varepsilon$ within $T = \mathrm{poly}(\frac{1}{\varepsilon}, \text{input size})$ steps.

Analysis: Smoothness implies F decreases in each step. Combine with a priori lower bound obtained using constructive invariant theory.

Corollary

Same algorithm solves null cone problem in time $\mathrm{poly}(\frac{1}{\gamma}, \text{input size})$.

Much faster (in theory and practice) than algebraic methods (e.g., via Gröbner bases).

Beyond gradient descent

There are more clever algorithms: trust region or interior-point methods, which we recently generalized to Hadamard manifolds [BFGOWW,Hirai-Nieuwboer-W].

These minimize local quadratic approximations

$$F(e^H g) \approx Q(H) = F(g) + \nabla F(g)[H] + \frac{1}{2}\nabla^2 F(g)[H, H]$$

on small neighborhoods. Need F "robust" or "self-concordant".

Theorem

If F bounded from below, algorithm finds $g \in G$ such that $F(g) \leqslant \inf_{g \in G} F(g) + \varepsilon$ within $T = \mathrm{poly}(\log \frac{1}{\varepsilon}, \text{input size}, \frac{1}{\gamma})$ steps.

State of the art: General algorithms for convex optimization in nonpositive curvature, which in particular solve problems in GIT. Polynomial time for many interesting actions – but not always!

Beyond gradient descent

There are more clever algorithms: trust region or interior-point methods, which we recently generalized to Hadamard manifolds [BFGOWW,Hirai-Nieuwboer-W].

These minimize local quadratic approximations

$$F(e^H g) \;\approx\; Q(H) = F(g) + \nabla F(g)[H] + \frac{1}{2}\nabla^2 F(g)[H, H]$$

on small neighborhoods. Need F "robust" or "self-concordant".

Theorem

If F bounded from below, algorithm finds $g \in G$ such that $F(g) \leqslant \inf_{g \in G} F(g) + \varepsilon$ within $T = \mathrm{poly}(\log \frac{1}{\varepsilon}, \text{input size}, \frac{1}{\gamma})$ steps.

State of the art: General algorithms for convex optimization in nonpositive curvature, which in particular solve problems in GIT. Polynomial time for many interesting actions – but not always!

Beyond gradient descent

There are more clever algorithms: trust region or interior-point methods, which we recently generalized to Hadamard manifolds [BFGOWW,Hirai-Nieuwboer-W].

These minimize local quadratic approximations

$$F(e^H g) \approx Q(H) = F(g) + \nabla F(g)[H] + \frac{1}{2}\nabla^2 F(g)[H, H]$$

on small neighborhoods. Need F "robust" or "self-concordant".

Theorem

If F bounded from below, algorithm finds $g \in G$ such that $F(g) \leqslant \inf_{g \in G} F(g) + \varepsilon$ within $T = \text{poly}(\log \frac{1}{\varepsilon}, \text{input size}, \frac{1}{\gamma})$ steps.

State of the art: General algorithms for convex optimization in nonpositive curvature, which in particular solve problems in GIT. Polynomial time for many interesting actions – but not always!

Algorithms for moment polytopes

$F(g) = \log \|g \cdot v\|$

Recall scaling problem: Find $g \in G$ such that $\nabla F(g) \approx 0$. Depending on action, this can mean *doubly stochastic* matrix, ... $\rightsquigarrow$ uniform marginals.

More generally, we can ask nonuniform scaling problem:

Given $\mathbf{p}$, find $g \in G$ s.th. $\nabla F(g) \approx \mathbf{p}$.

In other words, want to prescribe an arbitrary value for the moment map.

Moment polytope captures possible values if restrict to Weyl chamber:

State of the art: Algorithms discussed above can also solve nonuniform scaling problem and membership in moment polytopes. Polynomial in most parameters for many interesting actions – but not in the bitsize of $\mathbf{p}$!

Algorithms for moment polytopes

Recall scaling problem: Find $g \in G$ such that $\nabla F(g) \approx 0$. Depending on action, this can mean *doubly stochastic* matrix, ... $\rightsquigarrow$ uniform marginals.

More generally, we can ask nonuniform scaling problem:

> Given $\mathbf{p}$, find $g \in G$ s.th. $\nabla F(g) \approx \mathbf{p}$.

In other words, want to prescribe an arbitrary value for the moment map.

Moment polytope captures possible values if restrict to Weyl chamber:

$$\Delta(v) = \{\mathbf{p} \ : \ \nabla F(g) \approx \mathbf{p}\}$$

> **State of the art:** Algorithms discussed above can also solve nonuniform scaling problem and membership in moment polytopes. Polynomial in most parameters for many interesting actions – but not in the bitsize of $\mathbf{p}$!

Algorithms for moment polytopes

Recall scaling problem: Find $g \in G$ such that $\nabla F(g) \approx 0$. Depending on action, this can mean *doubly stochastic* matrix, ... $\rightsquigarrow$ uniform marginals.

More generally, we can ask nonuniform scaling problem:

Given $\mathbf{p}$, find $g \in G$ s.th. $\nabla F(g) \approx \mathbf{p}$.

In other words, want to prescribe an arbitrary value for the moment map.

Moment polytope captures possible values if restrict to Weyl chamber:

$$\Delta(v) = \{\mathbf{p} \,:\, \nabla F(g) \approx \mathbf{p}\}$$

State of the art: Algorithms discussed above can also solve nonuniform scaling problem and membership in moment polytopes. Polynomial in most parameters for many interesting actions – but not in the bitsize of $\mathbf{p}$!

Algorithms for moment polytopes

Recall scaling problem: Find $g \in G$ such that $\nabla F(g) \approx 0$. Depending on action, this can mean *doubly stochastic* matrix, $\ldots \rightsquigarrow$ uniform marginals.

More generally, we can ask nonuniform scaling problem:

Given $\mathbf{p}$, find $g \in G$ s.th. $\nabla F(g) \approx \mathbf{p}$.

In other words, want to prescribe an arbitrary value for the moment map.

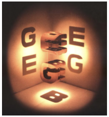
Moment polytope captures possible values if restrict to Weyl chamber:

$$\Delta(v) = \{\mathbf{p} \,:\, \nabla F(g) \approx \mathbf{p}\}$$

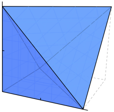
State of the art: Algorithms discussed above can also solve nonuniform scaling problem and membership in moment polytopes. Polynomial in most parameters for many interesting actions – but not in the bitsize of $\mathbf{p}$!

Summary and outlook

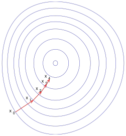
Symmetries lie behind many natural computational problems from algebra and analysis to classical and quantum CS.



Polytopes encode answers & clues to many of these problems. Often exp. vertices & facets, yet can admit efficient algos.



Symmetries are key to tackling these problems by optimization. Enabled by geodesic convexity and invariant theory.



Many open questions: Polynomial time algorithms for all actions? Complexity of algebraic problems? Structured or typical data? Other problems with natural symmetries? . . . **Happy birthday, Michèle!**