Operator scaling++

Yuanzhi Li (Princeton)

Joint work with Zeyuan Allen-Zhu, Ankit Garg(Microsoft Research), Rafael Oliveria (University of Toronto) and Avi Wigderson (IAS)

Thanks:

My fiancée Yandi Jin

Problem of interest

- Group action: A group G, a set V.
  - Each element g in G defines a map g: V -> V (actions).
  - The actions preserve the group structure:
    - $(g_1 g_2)(v) = g_1(g_2(v))$.
    - Identity(v) = v.

Problem of interest

- Orbit intersection
  - Given elements v_1, v_2 in V, whether there are g_1, g_2 in G such that
    - $g_1(v_1) = g_2(v_2)$.
- Orbit closure intersection:
  - Suppose V is some metric space.
  - Given elements v_1, v_2 in V, consider two sets:
    - $O(v_1) = \{ g(v_1) \mid g \text{ in } G\}$, $O(v_2) = \{ g(v_2) \mid g \text{ in } G\}$.
  - Whether the closures of $O(v_1)$ and $O(v_2)$ intersect.
- We want to do it efficiently.

Why is this problem interesting?

- Graph isomorphism:
  - G: Symmetric group on n elements.
  - V: Set of graphs with n vertices.

  - An efficient algorithm (poly time) would be a huge break through.

How hard is this problem?

- Orbit intersection: Unlikely to be NP-complete.
- ‘In co-AM’ just like Graph Isomorphism.

In this paper

- We give a polynomial time algorithm for orbit closure intersection problem for left-right linear actions:
  - V = set of m tuples of n x n complex matrices.
    - $v = (M_1, \ldots, M_m)$.
  - $G = SL_n(C) \times SL_n(C)$.
    - $SL_n(C)$: The set of all linear operations on C^n with determinate 1.
    - g = A, B.
  - $g(v) = (AM_1B^+, \ldots, AM_mB^+)$.

The main problem:

- Given $v_1 = (M_1, \dots, M_m)$, $v_2 = (M_1', \dots, M_m')$, decide whether for every œ > 0 there exists A, B in such that
  - $||AM_1B^+ - M_1'||_F^2 + \dots + ||AM_mB^+ - M_m'||_F^2 <$ œ .
- We give a polynomial time algorithm:
  - Polynomial in the number of bits to write down v_1, v_2.

Optimization approach

- Let us focus on $V = C^n$ (complex Euclidean space of dimension n), G is a subset of $GL_n(C)$.
- $|| * ||_2$ is the Euclidean norm.
  - Define $f_v(g) = ||g(v)||_2^2$.
  - Define $N(v) = \inf_{g \text{ in } G} f_v(g)$.
  - Moment map: $\mu_G(v)$.

Optimization approach

- Kempf-Ness Theorem + Hilbert Nullstellansatz:
  - Suppose $N(v_1)$, $N(v_2) > 0$, then there is an element in the intersection of the closures of $O(v_1)$ and $O(v_2)$:
  - Then there is one v_0 in the intersection of the closures of $O(v_1)$ and $O(v_2)$ such that:
    - $\mu_G(v_0) = 0$.
  - v_0 is unique in the sense that for every such v_0, v_0', there exists s with:
    - $v_0 = s(v_0')$, where:
      - $||s(v')||_2 = ||v'||_2$ for every v' in V.
        - s is in the maximum compact subgroup K of G.

Two steps approach

- Find $v_1' = g_1(v_1)$, $v_2' = g_2(v_2)$ such that $\mu_G(v_1') = \mu_G(v_2') = 0$.
  - Find g_1, g_2 that minimizes $||g_1(v_1)||_2^2$, $||g_2(v_2)||_2^2$.
    - Optimization: given v, find the argmin of $||g(v)||_2^2$ over g in G.
- Solve the problem whether the orbit of v_1', v_2' in K intersects
  - Solve the problem on a simpler group (The maximum compact subgroup).
- Original $\leftrightarrow$ Opt + Simple.

The idea looks simple

- But there is a problem:
  - The optimization step: Given v, find the argmin of $||g(v)||_2^2$ over g in G.
  - Usually, the theorems in optimization looks like this:
    - Given v, find g' such that $||g'(v)||_2^2 \leq \mathrm{Inf}_{g \text{ in } G} ||g(v)||_2^2 + \epsilon$.
      - In some Time($1/\epsilon$).
  - Two reasons:
    - Infimum might not be achievable.
    - Infimum might not be a rational matrix (even when v is an integer vector).

Can we work inexactly?

- One major difference between optimization and mathematics:
  - In math:
    - The exact minimizer has property blah blah blah.
    - If \$ equals to ¥, then blah blah blah.

  - In optimization:
    - To get an efficient algorithm, most of the time we need to work with:
      - The approximate minimizer.
      - When \$ approximately equals to ¥.

Two steps approach (Modified)

- Find g_1, g_2 that approximately minimizes $||g_1(v_1)||_2^2$, $||g_2(v_2)||_2^2$
  - Optimization: given v, approximately minimizes $||g(v)||_2^2$ over g in G.
    - $||g'(v)||_2^2 \leq \text{Inf}_{g \text{ in } G}||g(v)||_2^2 + \epsilon$.

- Solve the problem whether the orbit of v_1', v_2' in K approximately intersects:
  - Find s_1 , s_2 in K such that:
  - $||\ s_1(v_1') - s_2(v_2')\ ||_2^2 \leq \epsilon$.

- Original $\leftrightarrow$ Opt(ϵ) + Simple (ϵ)?

How to choose epsilon?

- How fast we can minimize the given function?
  - Given v, find g' such that $||g'(v)||_2^2 \le \text{Inf}_{g \text{ in } G} ||g(v)||_2^2 + \epsilon$
    - In some Time($1/\epsilon$).
  - What is this Time($1/\epsilon$)?
    - Logarithmic? $\log(1/\epsilon)$?
    - Polynomial? $(1/\epsilon)^{10}$?
    - Exponential? $e^{1/\epsilon}$?
  - → we want large epsilon so we can optimize fast.
- How small the error needs so the mathematical theorems still hold:
  - → we want small epsilon so we can prove the theorem easily.

How fast we can minimize the given function

- Given v, minimize $||g(v)||_2^2$ over g in G.
- Theorem of [GRS'13, Wood'11]
  - For 'most of the' G: a subset of $GL_n(C)$, $V = C^n$:
    - $||g(v)||_2^2$ is geodesically convex.

Geodesic Convexity

- Equip some Riemannian metric ||*|| on G: a subset of $GL_n(C)$.
  - Geodesic path from g_1 to g_2:
    - γ: [0, 1] -> G
    - $\gamma(0) = g_1$
    - $\gamma(1) = g_2$
    - For every s, t in [0,1]: || $\gamma(s) - \gamma(t)$ || is proportional to |s - t|.
  - It can also be (uniquely) characterized by $\gamma'(0)$.
- A function f: G -> R is geodesically convex:
  - If and only if for every geodesic path γ:
  - $f(\gamma)$: [0, 1] -> R is convex.

Minimize a geodesically convex function

- Convex function f:
  - Gradient descent: at each point x, move along $-\nabla f(x)$.

- Geodesic convex function f:
  - Geodesic gradient descent: At each point x, find geodesic path γ such that
  - $\gamma(0) = x$.
  - $\gamma'(0) = -\nabla f(x)$.
  - Move along γ.

Minimize a Geodesically convex function

- Theorem[ZS'16]:
  - Using Geodesic gradient descent, we can minimize a geodesic convex function f up to error ϵ in time:
    - $(1/\epsilon)^2$.
  - Polynomial time algorithm for (inverse) polynomially small ϵ.

Mathematical part

- Can we prove the theorem when the error ϵ is inverse polynomial?
  - It is ok for applications in previous work: [GGOW'16] (Null Cone)
  - Not ok for our problem: We need exponentially small epsilon.

Mathematical part

- The inexact orbit closure intersection problem:
  - Given tuples $v_1 = (M_1, \ldots, M_m)$ and $v_2 = (M_1', \ldots, M_m')$.
  - Our group G: $SL_n(C) \times SL_n(C)$.
  - The orbit closure of v_1 and v_2 intersects:
    - if and only if there exists (A, B) in G such that
    - $||AM_1B^+ - M_1'||_F^2 + \ldots + ||AM_mB^+ - M_m'||_F^2 \leq \epsilon$.
- Central question: How small ϵ needs to be?

Mathematical part

- Theorem (this paper): epsilon being (inverse) exponentially small is sufficient:
- There exists a polynomial p: R^3 -> R such that for every m, n, B, every $v_1 = (M_1, \ldots, M_m)$ and $v_2 = (M_1', \ldots, M_m')$ where M_i, M_i' in $C^{n \times n}$ with each entry being integers in [-B, B], the following two statements are equivalent:
  - The orbit closure of v_1 and v_2 intersects.
  - There exists (A, B) in G such that $||AM_1B^+ - M_1'||_F^2 + \ldots + ||AM_mB^+ - M_m'||_F^2 \leq e^{-p(m, n, \log B)}$.
- (inverse) exponential is tight.

Epsilon is inverse exponentially small

- To get an efficient algorithm, we need to:
  - Given v, find g' such that $||g'(v)||_2^2 \le \text{Inf}_{g \text{ in } G} ||g(v)||_2^2 + \epsilon$.
    - In time polylog(1/ϵ).
- Which means that we can not use gradient descent.

Faster optimization algorithms

- Optimization algorithm with polylog($1/\epsilon$) convergence rate:
  - In convex setting:
    - Newton's method (Only local convergence rate).
    - Interior point algorithm.
    - Ellipsoid algorithm.

Faster optimization algorithms:

- Newton’s method (Only local convergence rate).
- Interior point algorithm.
  - They are ‘gradient descent’ type of algorithms.
  - They all based on the so called ``self-concordant function’’.

Self-concordant function:

- A convex function f: R -> R is self-concordant if for every x:
  - $|f'''(x)| \leq 2\,(f''(x))^{3/2}$.
  - Scaling independent: $|f'''(ax)| \leq 2\,(f''(ax))^{3/2}$ for every non-zero a.

- A geodesically convex function is self-concordant if for every geodesic path γ, f(γ): R -> R is self-concordant.

Self-concordant function:

- A convex function f: R -> R is self-concordant if for every x:
  - $|f'''(x)| \leq 2\,(f''(x))^{3/2}$
- ‘looks likes’ Quadratic function:
  - When the second order derivative is small, the change of it is also small.

Optimize a self-concordant function:

- Forklore (informal): There is an algorithm that runs in time poly(n)log(1/ϵ) to minimize a geodesic self-concordant function: G -> R up to accuracy ϵ.
- Move along direction $-(\nabla^2 f(x))^{-1}\nabla f(x)$.

Good, can we use it?

- No…
  - Our function $||g(v)||_2^2$ is not geodesic self-concordant …

Ok, can we modify the definition?

- $|f'''(x)| \le 2\,(f''(x))^{3/2}$
- Why 3/2? I don't like 3/2… it's not an integer… not nice…
- Scaling independent.
- Can we fix a scaling and use other powers?

Self-robust function:

- A function f: R -> R is self-robust if for every x:
  - $|f'''(x)| \leq f''(x)$.
- A geodesic convex function is self-concordant if for every unit speed geodesic path γ, $f(\gamma)$: R -> R is self-concordant.
  - Unit speed: $||\gamma'(0)|| = 1$.

Optimize a self-robust function:

- Theorem [This paper, informal]: There is an algorithm that runs in time poly(n)log(1/ϵ) to minimize a geodesic self-robust function: G -> R up to accuracy ϵ. (G is a subset of $GL_n(C)$)

Overview of the algorithm:

- At every iteration, maintain a point g in G.
  - Compute the local geodesic gradient $\nabla f(g)$, defined as:
    - For every direction e, $\langle e, \nabla f(g) \rangle = f'(\gamma(t))|_{t=0}$
      - Such that $\gamma(0) = g$, $\gamma'(0) = e$.
  - Compute the local geodesic hessian $\nabla^2 f(g)$, defined as
    - For every direction e, $e^+ \nabla^2 f(g)\, e = f''(\gamma(t))|_{t=0}$
      - Such that $\gamma(0) = g$, $\gamma'(0) = v$.
  - Minimize the function $g(e) = \langle e, \nabla f(g) \rangle + 0.1\, e^+ \nabla^2 f(g)\, e$ over $||e|| \leq 0.1$
    - Let e^* be the minimizer.
    - Move to $\gamma(0.01)$ such that
      - $\gamma(0) = g$, $\gamma'(0) = e^*$.

Wait a second…

- Why $\nabla f(g)$, $\nabla^2 f(g)$ even exist?
- It is important that γ is geodesic path.
  - Can be defined via ‘Exponential map’.

Good, can we use it?

- No…
  - Our function $||g(v)||_2^2$ is not geodesic self-robust…

Wait… so what are you talking about???

Modify the function

- We just need to find the minimizer g in G of $||g(v)||_2^2$.
- We can minimize any function $h_v(g)$ such that
  - $\text{argmin}_{g \text{ in } G}\, h_v(g) = \text{argmin}_{g \text{ in } G} ||g(v)||_2^2$.
- Can we find such function? And make it self-concordant/self-robust?

The log capacity function:

- In our problem:
  - $v = (M_1, \ldots, M_m)$, $g = (A, B)$.
  - $||g(v)||_2^2 = ||AM_1B^+||_F^2 + \ldots + ||AM_mB^+||_F^2$.
  - Minimizing this function is aka operator scaling.
- The equivalent function: log capacity function: for a PSD matrix X,
  - $f(X) = \mathrm{logdet}(M_1XM_1^+ + \ldots + M_mXM_m^+) - \mathrm{logdet}(X)$.
- Theorem [Gurvits'04]:
  - Let X^* be a minimizer of $f(X)$, let A^*, B^* be the minimizer of $||g(v)||_2^2$, then there exists a, b in R such that
    - $B^+B = a X^*$.
    - $AA^+ = b(M_1XM_1^+ + \ldots + M_mXM_m^+)$.

Log capacity function:

- Theorem [This paper]:
- $f(X) = \text{logdet}(M_1XM_1^+ + \ldots + M_mXM_m^+) - \text{logdet}(X)$
  - Is a geodesic self-robust function, with the geodesic path over PSD matrices given as:
  - $\gamma(t) = X_0^{1/2}e^{tD}X_0^{1/2}$
    - $\gamma(0) = X_0$
    - $\gamma'(0) = X_0^{1/2}DX_0^{1/2}$.

Step 1:

- Minimize:
  - $f_1(X) = \text{logdet}(M_1 X M_1^{+} + \ldots + M_m X M_m^{+}) - \text{logdet}(X)$.
  - $f_2(X) = \text{logdet}(M_1' X M_1'^{+} + \ldots + M_m' X M_m'^{+}) - \text{logdet}(X)$.
  - Let X_1^*, X_2^* be an ϵ-approximate minimizers.
- Let:
  - $B^+B = a_1 X_1^*$, $B_2^+B_2 = a_2 X_2^*$.
  - $AA^+ = b_1(M_1 X_1^* M_1^{+} + \ldots + M_m X_1^* M_m^{+})$, $A_2 A_2^+ = b_2(M_1' X_2^* M_1'^{+} + \ldots + M_m' X_2^* M_m'^{+})$.

Step 2:

- Now, let $w_1 = Av_1B^+$, $w_2 = A_2v_2B_2^+$.
- Check the orbit closure $w_{1,}$ w_2 approximately interests in a subgroup K:
  - K: all the elements g in G such that for every v in V, $||g(v)||_2^2 = ||v||_2^2$
  - In this problem: K: the set of all (determinate one) unitary matrices.

Exact unitary equivalence:

- Given $w_1 = (M_1, \ldots, M_m)$, $w_2 = (M_1', \ldots, M_m')$,
- Check whether there exists unitary matrices U, V such that
  - For all i in m: $U M_i V^+ = M_i'$.
- Existing algorithms [CIK'97, IQ'18] can solve it in polynomial time.

Inexact unitary equivalence:

- Given $w_1 = (M_1, \ldots, M_m)$, $w_2 = (M_1', \ldots, M_m')$,
- Check whether there exists unitary matrices U, V such that
  - For all i in m: $||U M_i V^+ - M_i'||_F \leq \epsilon$.
    - Where ϵ is (inverse) exponentially small.

Naïve idea

- Given $w_1 = (M_1, \ldots, M_m)$, $w_2 = (M_1', \ldots, M_m')$,
- Check whether there exists unitary matrices U, V such that
  - For all i in m: $||U\, M_i\, V^+ - M_i'||_F \leq \epsilon$.
    - Where ϵ is (inverse) exponentially small.
  - Make ϵ smaller than (inverse) exponential of the bit complexity of M_i and M_i'.
  - So $U\, M_i\, V^+ = M_i'$.
    - Use exact algorithm.

Recall how we get w_1

- Minimize:
  - $f_1(X) = \text{logdet}(M_1 X M_1^+ + \ldots + M_m X M_m^+) - \text{logdet}(X)$.
  - $f_2(X) = \text{logdet}(M_1' X M_1'^+ + \ldots + M_m' X M_m'^+) - \text{logdet}(X)$.
  - Let X_1^*, X_2^* be an ϵ-approximate minimizers.
- Let:
  - $B^+B = a_1 X_1^*$, $B_2^+B_2 = a_2 X_2^*$.
  - $AA^+ = b_1(M_1 X_1^* M_1^+ + \ldots + M_m X_1^* M_m^+)$, $A_2A_2^+ = b_2(M_1' X_2^* M_1'^+ + \ldots + M_m' X_2^* M_m'^+)$.

Naïve idea

- Given $w_1 = (M_1, \ldots, M_m)$, $w_2 = (M_1', \ldots, M_m')$:
  - $w_1 = Av_1B^+$, where A, B defined by an ϵ-approximate minimizer.
- The smaller ϵ is, the larger the bit complexity of M_i is.
  - In fact, ϵ can never be smaller than (inverse) exponential of the bit complexity of M_i and M_i'.

This paper

- We give an algorithm that runs in time poly(m, n, log 1/ϵ, B) that
  - Given $w_1 = (M_1, \ldots, M_m)$, $w_2 = (M_1', \ldots, M_m')$,
  - Distinguish whether there exists unitary matrices U, V such that
    - For all i in m: $||U M_i V^+ - M_i'||_F \leq \epsilon$.
    - There exists i in m: $||U M_i V^+ - M_i'||_F \geq \exp(\text{poly}(m, n))\epsilon^{1/\text{poly}(m, n)}$.

- It applies to any ϵ, regardless of the bit complexity B of w_1, w_2.

In exact algorithm: Step 1

- What do we know if there exists unitary U, V such that
  - $||U\, M\, V^{+} – M'||_F \leq \epsilon$.
- Singular value decomposition.

Singular value decomposition (SVD)

- For every matrix M in $C^{n \times n}$:
- There exists unitary matrices U, V and a diagonal matrix $\Sigma = \mathrm{diag}(\sigma_1, \ldots, \sigma_n)$ with $\sigma_1 \geq \sigma_2 \geq \ldots \geq \sigma_n \geq 0$ such that:
  - $M = U\Sigma V^+$.

Gap-free Wedin Theorem

- What do we know if there exists unitary U, V such that
  - $||U\, M\, V^+ - M'||_F \leq \epsilon$?
- Theorem [AL'16]: Let $M = U_1 \Sigma V_1^+$, $M' = U_2 \Sigma V_2^+$ be the SVD of M, M'. Suppose $||U\, M\, V^+ - M'||_F \leq \epsilon$. For every $\delta \geq 0$, if
  - There exists k in [n - 1] such that $\sigma_k - \sigma_{k+1} \geq \delta$
    - Then $||U_{11}{}^+ U^+ U_{22}||_F \leq \epsilon/(\delta - 2\epsilon)$.
      - U_{11}: First k columns of U_1, U_{22}: Last n – k columns of U_2

Gap-free Wedin Theorem

- $||U_{11}{}^{+} U^{+}U_{22}||_F \leq \epsilon/(\delta - 2\epsilon)$.
- If we write $U = U_2U'U_1{}^{+}$ for an unitary matrix U'.
  - Then U' is close to being (block) diagonal.

Apply gap-free Wedin Theorem

- Suppose $||U M_i V^+ - M_i'||_F \leq \epsilon$ and M_i has a singular value gap δ
  - Then we can find in polynomial time unitary U’, U’’, V’, V’’ such that there exists unitary matrices U_1, U_2, V_1, V_2:
    - $|| U - U' \operatorname{diag}(U_1, U_2) U''^+ ||_F \leq \epsilon/(\delta - 2\epsilon)$.
    - $|| V - V' \operatorname{diag}(V_1, V_2) V''^+ ||_F \leq \epsilon/(\delta - 2\epsilon)$.
  - Reduce the original problem to two sub problems of smaller dimensions.

In the end:

- Left with a problem where all M_i has no singular value gap:
  - M_i is close to a (constant multiple of) unitary matrix.
- $||U M_i V^+ - M_i'||_F \leq \epsilon$ where M_i, M_i' are close to unitary:
  - V is close to $M_i'^+ U M$.
- So we can reduce one unitary V and focus only on U.

Inexact unitary conjugation:

- Given $w_1 = (M_1, \ldots, M_m)$, $w_2 = (M_1', \ldots, M_m')$,
- Check whether there exists unitary matrices U such that
  - For all i in m: $||U M_i U^+ - M_i'||_F \leq \epsilon$.
    - Where ϵ is (inverse) exponentially small.

Eigenvalue

- The eigenvalues of a matrix M is given by the set of all values λ in C such that
  - Det(λI - M) = 0.

Eigenvalue Wedin Theorem

- What do we know if there exists unitary U such that
  - $||U M U^{+} – M'||_F \leq \epsilon$.
- Theorem [This paper]: Suppose $||U M U^{+} – M'||_F \leq \epsilon$, then for every $\delta \geq 0$, if M has two eigenvalues λ_1, λ_2 such that $|\lambda_1 - \lambda_2| \geq \delta$,
  - We can compute, in time poly(n, log1/ϵ) unitary matrices U', U'' such that there exists unitary matrices U_1, U_2:
    - $|| U - U' \mathrm{diag}(U_1, U_2) U''^{+}||_F \leq \epsilon/(\delta/n - 2\epsilon)$.

Apply eigenvalue Wedin Theorem

- Suppose $||U\, M_i\, U^+ - M_i'||_F \le \epsilon$ and M_i has a eigenvalue gap δ:
  - Then we can find in polynomial time unitary matrices U' ,U'' such that there exists unitary matrices U_1, U_2
    - $||\, U - U' \,\mathrm{diag}(U_1, U_2)\, U''^+ \,||_F \le \epsilon/(\delta/n - 2\epsilon)$.

  - Reduce the original problem to two sub problems of smaller dimensions.

In the end:

- Left with a problem where all M_i has no singular value gap nor eigenvalue gap.
  - Theorem [This paper]: M_i is close to (constant multiple of) identity.

Summary

- We give a polynomial time algorithm for orbit closure intersection problem for left-right linear actions:
  - V = set of m tuples of n x n complex matrices.
    - $v = (M_1, \ldots, M_m)$.
  - $G = SL_n(C) \times SL_n(C)$.
    - g = A, B.
  - $g(v) = (AM_1B^+, \ldots, AM_mB^+)$.

Summary

- Use geodesic optimization to reduce the original problem to inexact unitary equivalence problem.
  - Inexact theorem holds for (inverse) exponentially small epsilon.
    - Mathematics.
  - Design an algorithm with linear convergence rate.
    - Optimization.
- Design a new algorithm for inexact unitary equivalence problem.