Operator Scaling with Specified Marginals

Cole Franks

June 26, 2018

Rutgers University

Organization

1. Examples; teaser algorithm
2. What is operator scaling?
3. Algorithmic results
4. Proofs

Special cases of operator scaling

1. Matrix scaling: Given nonnegative matrix A, can $D_1 A D_2$ have uniform row and column sums?
2. Isotropic position: Given distribution on vectors, can one change basis + *normalize* and end up in isotropic position ($\mathbb{E}[uu^\dagger] = I$)?
3. Schur-Horn theorem: what can arise as the diagonal of a matrix with given eigenvalues?
4. Horn's problem: given lists of numbers $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, are there PSD matrices $A + B + C = I$ with lists of eigenvalues $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, resp?
5. Operator scaling

Special cases of operator scaling

1. Matrix scaling: Given nonnegative matrix A, can D_1AD_2 have uniform row and column sums?
2. Isotropic position: Given distribution on vectors, can one change basis + *normalize* and end up in isotropic position ($\mathbb{E}[uu^\dagger] = I$)?
3. Schur-Horn theorem: what can arise as the diagonal of a matrix with given eigenvalues?
4. Horn's problem: given lists of numbers $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, are there PSD matrices $A + B + C = I$ with lists of eigenvalues $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, resp?
5. Operator scaling

Special cases of operator scaling

1. Matrix scaling: Given nonnegative matrix A, can $D_1 A D_2$ have uniform row and column sums?
2. Isotropic position: Given distribution on vectors, can one change basis + *normalize* and end up in isotropic position ($\mathbb{E}[uu^\dagger] = I$)?
3. Schur-Horn theorem: what can arise as the diagonal of a matrix with given eigenvalues?
4. Horn's problem: given lists of numbers $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, are there PSD matrices $A + B + C = I$ with lists of eigenvalues $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, resp?
5. Operator scaling

Special cases of operator scaling

1. Matrix scaling: Given nonnegative matrix A, can $D_1 A D_2$ have uniform row and column sums?
2. Isotropic position: Given distribution on vectors, can one change basis + *normalize* and end up in isotropic position ($\mathbb{E}[uu^\dagger] = I$)?
3. Schur-Horn theorem: what can arise as the diagonal of a matrix with given eigenvalues?
4. Horn's problem: given lists of numbers $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, are there PSD matrices $A + B + C = I$ with lists of eigenvalues $\boldsymbol{a}, \boldsymbol{b}, \boldsymbol{c}$, resp?
5. Operator scaling

Teaser algorithm: Horn's problem.

Decision algorithm MNS12 relies on difficult theorems KT99, BZ01.

Simple $O(bm/\varepsilon^2)$ time search algorithm!

Algorithm (F18)

Input: $\varepsilon > 0$, and three nonnegative diagonal matrices D_1, D_2, D_3.

Output: Orthogonal matrices O_1, O_2, O_3 with

$$\|O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger - I_m\| \leq \varepsilon$$

- Pick entries of O_i randomly from $[2^{2b}]$, b input size.
- Set $O_i \leftarrow O_i h_i$ orthogonal by *right* multiplication with upper trangulars h_i.
- Check if done. Else, enforce $O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger = I_m$ by simultaneous *left* multiplication of $O_i \leftarrow g O_i$ with a *lower* triangular g. *Go to 2.*

Teaser algorithm: Horn's problem.

Decision algorithm MNS12 relies on difficult theorems KT99, BZ01.

Simple $O(bm/\varepsilon^2)$ time search algorithm!

Algorithm (F18)

Input: $\varepsilon > 0$, and three nonnegative diagonal matrices D_1, D_2, D_3.

Output: Orthogonal matrices O_1, O_2, O_3 with

$$\|O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger - I_m\| \leq \varepsilon$$

- Pick entries of O_i randomly from $[2^{2b}]$, b input size.
- Set $O_i \leftarrow O_i h_i$ orthogonal by *right* multiplication with upper trangulars h_i.
- Check if done. Else, enforce $O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger = I_m$ by simultaneous *left* multiplication of $O_i \leftarrow g O_i$ with a *lower* triangular g. *Go to 2.*

Teaser algorithm: Horn's problem.

Decision algorithm MNS12 relies on difficult theorems KT99, BZ01.

Simple $O(bm/\varepsilon^2)$ time search algorithm!

Algorithm (F18)

Input: $\varepsilon > 0$, and three nonnegative diagonal matrices D_1, D_2, D_3.

Output: Orthogonal matrices O_1, O_2, O_3 with

$$\|O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger - I_m\| \leq \varepsilon$$

- Pick entries of O_i randomly from $[2^{2b}]$, b input size.
- Set $O_i \leftarrow O_i h_i$ orthogonal by *right* multiplication with upper trangulars h_i.
- Check if done. Else, enforce $O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger = I_m$ by simultaneous *left* multiplication of $O_i \leftarrow g O_i$ with a *lower* triangular g. *Go to 2.*

Teaser algorithm: Horn's problem.

Decision algorithm MNS12 relies on difficult theorems KT99, BZ01.

Simple $O(bm/\varepsilon^2)$ time search algorithm!

Algorithm (F18)

Input: $\varepsilon > 0$, and three nonnegative diagonal matrices D_1, D_2, D_3.

Output: Orthogonal matrices O_1, O_2, O_3 with

$$\|O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger - I_m\| \leq \varepsilon$$

- Pick entries of O_i randomly from $[2^{2b}]$, b input size.
- Set $O_i \leftarrow O_i h_i$ orthogonal by *right* multiplication with upper trangulars h_i.
- Check if done. Else, enforce $O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger = I_m$ by simultaneous *left* multiplication of $O_i \leftarrow g O_i$ with a *lower* triangular g. *Go to 2.*

Teaser algorithm: Horn's problem.

Decision algorithm MNS12 relies on difficult theorems KT99, BZ01.

Simple $O(bm/\varepsilon^2)$ time search algorithm!

Algorithm (F18)

Input: $\varepsilon > 0$, and three nonnegative diagonal matrices D_1, D_2, D_3.

Output: Orthogonal matrices O_1, O_2, O_3 with

$$\|O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger - I_m\| \leq \varepsilon$$

- Pick entries of O_i randomly from $[2^{2b}]$, b input size.
- Set $O_i \leftarrow O_i h_i$ orthogonal by *right* multiplication with upper trangulars h_i.
- Check if done. Else, enforce $O_1 D_1 O_1^\dagger + O_2 D_2 O_2^\dagger + O_3 D_3 O_3^\dagger = I_m$ by simultaneous *left* multiplication of $O_i \leftarrow g O_i$ with a *lower* triangular g. *Go to 2.*

Operator scaling

General framework: Operator scaling

Definition

An "operator" $\boldsymbol{A}$ is a tuple $A_1, \ldots, A_r$ of $m \times n$ matrices, and

$$\sum_{i=1}^{r} A_i A_i^{\dagger}, \qquad \sum_{i=1}^{r} A_i^{\dagger} A_i.$$

are the *left and right marginals* (resp.) of $\boldsymbol{A}$.

These are marginals of the two-body, mixed quantum state

$$\sum_{i=1}^{r} |A_i\rangle \langle A_i| .$$

General framework: Operator scaling

Definition

An "operator" $\boldsymbol{A}$ is a tuple $A_1, \ldots, A_r$ of $m \times n$ matrices, and

$$\sum_{i=1}^{r} A_i A_i^{\dagger}, \qquad \sum_{i=1}^{r} A_i^{\dagger} A_i.$$

are the *left and right marginals* (resp.) of $\boldsymbol{A}$.

These are marginals of the two-body, mixed quantum state

$$\sum_{i=1}^{r} |A_i\rangle \langle A_i| \,.$$

Scaling

Definition (scaling)

A scaling of a tuple $\boldsymbol{A}$ of matrices is another tuple obtained by simultaneously changing basis on the left and right, i.e. $A_i \leftarrow gA_ih$.

Question:

Given a tuple $\boldsymbol{A}$ and positive semidefinite matrices $P \succeq 0$, $Q \succeq 0$, Are there scalings of $\boldsymbol{A}$ with marginals arbitrarily close to P, Q?

If so, say $\boldsymbol{A}, P, Q$ is *scalable*.

May take P, Q diagonal w.l.o.g.

Scaling

Definition (scaling)

A scaling of a tuple $\boldsymbol{A}$ of matrices is another tuple obtained by simultaneously changing basis on the left and right, i.e. $A_i \leftarrow g A_i h$.

Question:

Given a tuple $\boldsymbol{A}$ and positive semidefinite matrices $P \succeq 0$, $Q \succeq 0$, Are there scalings of $\boldsymbol{A}$ with marginals arbitrarily close to P, Q?

If so, say $\boldsymbol{A}, P, Q$ is *scalable*.

May take P, Q diagonal w.l.o.g.

Scaling

Definition (scaling)

A scaling of a tuple $\boldsymbol{A}$ of matrices is another tuple obtained by simultaneously changing basis on the left and right, i.e. $A_i \leftarrow gA_ih$.

Question:

Given a tuple $\boldsymbol{A}$ and positive semidefinite matrices $P \succeq 0$, $Q \succeq 0$, Are there scalings of $\boldsymbol{A}$ with marginals arbitrarily close to P, Q?

If so, say $\boldsymbol{A}, P, Q$ is *scalable*.

May take P, Q diagonal w.l.o.g.

Some prior work

Producing ε-scalings on scalable instances.

- (HM13) P diagonal, $Q = I_m$, $A_i = \begin{bmatrix} 0 & \dots & | & \dots & 0 \\ & & u_i & & \\ & & | & & \end{bmatrix}$
  Isotropic position; certify no hidden subspace (SV17) $\operatorname{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = I_n, Q = I_n$; $\boldsymbol{A}$ arbitrary.
  Used for poly-time noncommutative rational identity testing.
  (AGLOW17) $\operatorname{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = \bigoplus p_i I_{n_i}$, $Q = I_n$, $A_i = \begin{bmatrix} 0 & \dots & \| & \dots & 0 \\ & & B_i & & \\ & & \| & & \end{bmatrix}$
  used to approximate Brascamp-Lieb constant from analysis.

Some prior work

Producing ε-scalings on scalable instances.

- (HM13) P diagonal, $Q = I_m$, $A_i = \begin{bmatrix} 0 & \dots & | \\ & & u_i \\ & & | \end{bmatrix} \dots \; 0$

$$A_i = \left[\begin{array}{ccccc} 0 & \dots & \begin{array}{c} | \\ u_i \\ | \end{array} & \dots & 0 \end{array} \right]$$

  Isotropic position; certify no hidden subspace (SV17) poly $\log(1/\varepsilon)$.
- (GGOW16) $P = I_n, Q = I_n$; $\boldsymbol{A}$ arbitrary.
  Used for poly-time noncommutative rational identity testing.
  (AGLOW17) poly $\log(1/\varepsilon)$.
- (GGOW16) $P = \bigoplus p_i I_{n_i}$, $Q = I_n$, $A_i = \left[\begin{array}{ccccc} 0 & \dots & \begin{array}{c} \| \\ B_i \\ \| \end{array} & \dots & 0 \end{array} \right]$
  used to approximate Brascamp-Lieb constant from analysis.

Some prior work

Producing ε-scalings on scalable instances.

- (HM13) P diagonal, $Q = I_m$, $A_i = \begin{bmatrix} 0 & \dots & u_i & \dots & 0 \end{bmatrix}$

  Isotropic position; certify no hidden subspace (SV17) $\text{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = I_n, Q = I_n$; $\boldsymbol{A}$ arbitrary.

  Used for poly-time noncommutative rational identity testing.

  (AGLOW17) $\text{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = \bigoplus p_i I_{n_i}$, $Q = I_n$, $A_i = \begin{bmatrix} 0 & \dots & B_i & \dots & 0 \end{bmatrix}$

  used to approximate Brascamp-Lieb constant from analysis.

Some prior work

Producing ε-scalings on scalable instances.

- (HM13) P diagonal, $Q = I_m$, $A_i = \begin{bmatrix} 0 & \dots & u_i & \dots & 0 \end{bmatrix}$
  Isotropic position; certify no hidden subspace (SV17) $\text{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = I_n, Q = I_n$; $\boldsymbol{A}$ arbitrary.
  Used for poly-time noncommutative rational identity testing.
  (AGLOW17) $\text{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = \bigoplus p_i I_{n_i}$, $Q = I_n$, $A_i = \begin{bmatrix} 0 & \dots & B_i & \dots & 0 \end{bmatrix}$
  used to approximate Brascamp-Lieb constant from analysis.

Some prior work

Producing ε-scalings on scalable instances.

- (HM13) P diagonal, $Q = I_m$, $A_i = \begin{bmatrix} 0 & \ldots & u_i & \ldots & 0 \end{bmatrix}$
  Isotropic position; certify no hidden subspace (SV17) $\text{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = I_n, Q = I_n$; $\boldsymbol{A}$ arbitrary.
  Used for poly-time noncommutative rational identity testing.
  (AGLOW17) $\text{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = \bigoplus p_i I_{n_i}$, $Q = I_n$, $A_i = \begin{bmatrix} 0 & \ldots & B_i & \ldots & 0 \end{bmatrix}$
  used to approximate Brascamp-Lieb constant from analysis.

Some prior work

Producing ε-scalings on scalable instances.

- (HM13) P diagonal, $Q = I_m$, $A_i = \begin{bmatrix} 0 & \dots & \vert \\ & & u_i \\ & & \vert \end{bmatrix}$ with $\dots\ 0$, i.e. $A_i = \begin{bmatrix} 0 & \dots & u_i & \dots & 0 \end{bmatrix}$

  Isotropic position; certify no hidden subspace (SV17) poly $\log(1/\varepsilon)$.
- (GGOW16) $P = I_n, Q = I_n$; $\boldsymbol{A}$ arbitrary.

  Used for poly-time noncommutative rational identity testing.

  (AGLOW17) poly $\log(1/\varepsilon)$.
- (GGOW16) $P = \bigoplus p_i I_{n_i}$, $Q = I_n$, $A_i = \begin{bmatrix} 0 & \dots & B_i & \dots & 0 \end{bmatrix}$

  used to approximate Brascamp-Lieb constant from analysis.

Some prior work

Producing ε-scalings on scalable instances.

- (HM13) P diagonal, $Q = I_m$, $A_i = \begin{bmatrix} 0 & \dots & \begin{matrix} | \\ u_i \\ | \end{matrix} & \dots & 0 \end{bmatrix}$
  Isotropic position; certify no hidden subspace (SV17) $\operatorname{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = I_n, Q = I_n$; $\boldsymbol{A}$ arbitrary.
  Used for poly-time noncommutative rational identity testing.
  (AGLOW17) $\operatorname{poly}\log(1/\varepsilon)$.
- (GGOW16) $P = \bigoplus p_i I_{n_i}$, $Q = I_n$, $A_i = \begin{bmatrix} 0 & \dots & \begin{matrix} || \\ B_i \\ || \end{matrix} & \dots & 0 \end{bmatrix}$
  used to approximate Brascamp-Lieb constant from analysis.

This work: A arbitrary, P, Q arbitrary.

Theorem (F18)

Exists a randomized algorithm to find ε-scalings of scalable $\boldsymbol{A}$, P, Q

$$O(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$$

where b is the bit complexity of the input.

Later (BFGOWW18): $\text{poly}(b, 1/\varepsilon)$ for one-body quantum marginals

This work: A arbitrary, P, Q arbitrary.

Theorem (F18)

Exists a randomized algorithm to find ε-scalings of scalable $\boldsymbol{A}$, P, Q

$$O(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$$

where b is the bit complexity of the input.

Later (BFGOWW18): $\mathrm{poly}(b, 1/\varepsilon)$ for one-body quantum marginals

The algorithms

Old scaling algorithm

The following algorithm works in all the previously solved cases:

Algorithm (Naïve scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q.

- Check if done. Else, left-scale $\boldsymbol{A} \leftarrow g\boldsymbol{A}$ so that $\sum_{i=1}^{r} A_i P A_i^\dagger = I$.
- Check if done. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ so that $\sum_{i=1}^{r} A_i^\dagger Q A_i = I$.

One problem - *many choices* for how to scale; each matters!

In previous success stories, scalings commuted with P, Q, so choice didn't affect later steps. Suggests *careful choice necessary*.

Old scaling algorithm

The following algorithm works in all the previously solved cases:

Algorithm (Naïve scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q.

- Check if done. Else, left-scale $\boldsymbol{A} \leftarrow g\boldsymbol{A}$ so that $\sum_{i=1}^{r} A_i P A_i^\dagger = I$.
- Check if done. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ so that $\sum_{i=1}^{r} A_i^\dagger Q A_i = I$.

One problem - *many choices* for how to scale; each matters!

In previous success stories, scalings commuted with P, Q, so choice didn't affect later steps. Suggests *careful choice necessary*.

Old scaling algorithm

The following algorithm works in all the previously solved cases:

Algorithm (Naïve scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q.

- Check if done. Else, left-scale $\boldsymbol{A} \leftarrow g\boldsymbol{A}$ so that $\sum_{i=1}^{r} A_i P A_i^\dagger = I$.
- Check if done. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ so that $\sum_{i=1}^{r} A_i^\dagger Q A_i = I$.

One problem - *many choices* for how to scale; each matters!

In previous success stories, scalings commuted with P, Q, so choice didn't affect later steps. Suggests *careful choice necessary*.

Old scaling algorithm

The following algorithm works in all the previously solved cases:

Algorithm (Naïve scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q.

- Check if done. Else, left-scale $\boldsymbol{A} \leftarrow g\boldsymbol{A}$ so that $\sum_{i=1}^{r} A_i P A_i^\dagger = I$.
- Check if done. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ so that $\sum_{i=1}^{r} A_i^\dagger Q A_i = I$.

One problem - *many choices* for how to scale; each matters!

In previous success stories, scalings commuted with P, Q, so choice didn't affect later steps. Suggests *careful choice necessary*.

Old scaling algorithm

The following algorithm works in all the previously solved cases:

Algorithm (Naïve scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q.

- Check if done. Else, left-scale $\boldsymbol{A} \leftarrow g\boldsymbol{A}$ so that $\sum_{i=1}^{r} A_i P A_i^\dagger = I$.
- Check if done. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ so that $\sum_{i=1}^{r} A_i^\dagger Q A_i = I$.

One problem - *many choices* for how to scale; each matters!
In previous success stories, scalings commuted with P, Q, so choice didn't affect later steps. Suggests *careful choice necessary*.

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^\dagger \boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$, *Random+Triangular outputs ERROR with probability at most* $1/3$ *if* $\boldsymbol{A}, P, Q$ *is scalable.*

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^{\dagger}\boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$, Random+Triangular outputs ERROR with probability at most $1/3$ if $\boldsymbol{A}, P, Q$ is scalable.

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^{\dagger}\boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$*, Random+Triangular outputs ERROR with probability at most* $1/3$ *if* $\boldsymbol{A}, P, Q$ *is scalable.*

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^{\dagger}\boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$, Random+Triangular outputs ERROR with probability at most $1/3$ if $\boldsymbol{A}, P, Q$ is scalable.

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^{\dagger}\boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$, *Random+Triangular outputs ERROR with probability at most* $1/3$ *if* $\boldsymbol{A}, P, Q$ *is scalable.*

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^{\dagger}\boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$, Random+Triangular outputs ERROR with probability at most $1/3$ if $\boldsymbol{A}, P, Q$ is scalable.

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^{\dagger}\boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$, Random+Triangular outputs ERROR with probability at most $1/3$ if $\boldsymbol{A}, P, Q$ is scalable.

New scaling algorithm

A good choice: random initial and all the rest upper triangular.

Algorithm (Random + Triangular)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to marginals P, Q or ERROR.

- Scale $\boldsymbol{A}$ on left and right by *random matrices* with entries in $[6 \cdot 2^{2b}]$.
- Repeat, t times returning ERROR if not possible:
  - If done, **return** $\boldsymbol{A}$. Else, left-scale $\boldsymbol{A} \leftarrow g^{\dagger}\boldsymbol{A}$ with g *upper triangular*.
  - If done, **return** $\boldsymbol{A}$. Else, right-scale $\boldsymbol{A} \leftarrow \boldsymbol{A}h$ with h *upper triangular*.
- return ERROR

Theorem

Provided $t = \Omega(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$, *Random+Triangular outputs ERROR with probability at most* $1/3$ *if* $\boldsymbol{A}, P, Q$ *is scalable.*

The proofs

Proof idea: reduction

Imagine we had a map $\boldsymbol{A} \mapsto \tilde{\boldsymbol{A}}$ so that

scaling steps on $\boldsymbol{A}$

$\leftrightarrow$ naïve scaling steps on $\tilde{\boldsymbol{A}}$, which converge! (Gu04)

$$\boldsymbol{A} \text{ has left marginal } P \iff \tilde{\boldsymbol{A}} \text{ has left marginal } I$$

$$\boldsymbol{A} \text{ has right marginal } Q \iff \tilde{\boldsymbol{A}} \text{ has right marginal } I.$$

and

Proof idea: reduction

Imagine we had a map $\boldsymbol{A} \mapsto \tilde{\boldsymbol{A}}$ so that

scaling steps on $\boldsymbol{A}$

$\leftrightarrow$ naïve scaling steps on $\tilde{\boldsymbol{A}}$, which converge! (Gu04)

$$\boldsymbol{A} \text{ has left marginal } P \iff \tilde{\boldsymbol{A}} \text{ has left marginal } I$$

$$\boldsymbol{A} \text{ has right marginal } Q \iff \tilde{\boldsymbol{A}} \text{ has right marginal } I.$$

and

Proof idea: reduction

Imagine we had a map $\boldsymbol{A} \mapsto \tilde{\boldsymbol{A}}$ so that

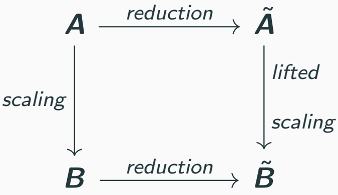
scaling steps on $\boldsymbol{A}$

$\leftrightarrow$ naïve scaling steps on $\tilde{\boldsymbol{A}}$, which converge! (Gu04)

$$\boldsymbol{A} \text{ has left marginal } P \iff \tilde{\boldsymbol{A}} \text{ has left marginal } I$$

$$\boldsymbol{A} \text{ has right marginal } Q \iff \tilde{\boldsymbol{A}} \text{ has right marginal } I.$$

and



Proof idea: reduction

Imagine we had a map $\boldsymbol{A} \mapsto \tilde{\boldsymbol{A}}$ so that

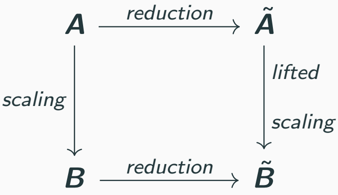
scaling steps on $\boldsymbol{A}$

$\leftrightarrow$ naïve scaling steps on $\tilde{\boldsymbol{A}}$, which converge! (Gu04)

$$\boldsymbol{A} \text{ has left marginal } P \iff \tilde{\boldsymbol{A}} \text{ has left marginal } I$$

$$\boldsymbol{A} \text{ has right marginal } Q \iff \tilde{\boldsymbol{A}} \text{ has right marginal } I.$$

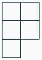
and

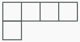


Reduction example

Reduction for

$$P = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } Q = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}.$$



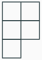


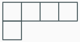
$$\underset{A_i}{\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}} \mapsto \underset{\tilde{A}_{i1}}{\begin{bmatrix} a & b & c & 0 & 0 \\ d & e & f & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}}, \underset{\tilde{A}_{i2}}{\begin{bmatrix} 0 & 0 & 0 & a & b \\ 0 & 0 & 0 & d & e \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{bmatrix}} \cdots$$

Reduction example

Reduction for

$$P = \begin{bmatrix} 2 & 0 & 0 \\ 0 & 2 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{ and } Q = \begin{bmatrix} 4 & 0 \\ 0 & 1 \end{bmatrix}.$$





$$\underbrace{\begin{bmatrix} a & b & c \\ d & e & f \end{bmatrix}}_{A_i} \mapsto \underbrace{\left[\begin{array}{ccc|cc} a & b & c & 0 & 0 \\ d & e & f & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}\right]}_{\tilde{A}_{i1}}, \underbrace{\left[\begin{array}{ccc|cc} 0 & 0 & 0 & a & b \\ 0 & 0 & 0 & d & e \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \\ \hline 0 & 0 & 0 & 0 & 0 \end{array}\right]}_{\tilde{A}_{i2}} \dots$$

Upper triangular scalings "lift"

Scaling $\boldsymbol{A}$ on right by

$$\begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix}$$

scales $\tilde{\boldsymbol{A}}$ on right by

$$\left[\begin{array}{ccc|cc} a & b & c & 0 & 0 \\ 0 & d & e & 0 & 0 \\ 0 & 0 & f & 0 & 0 \\ \hline 0 & 0 & 0 & a & b \\ 0 & 0 & 0 & 0 & d \end{array}\right]$$

Upper triangular scalings "lift"

Scaling $\boldsymbol{A}$ on right by

$$\begin{bmatrix} a & b & c \\ 0 & d & e \\ 0 & 0 & f \end{bmatrix}$$

scales $\tilde{\boldsymbol{A}}$ on right by

$$\left[\begin{array}{ccc|cc} a & b & c & 0 & 0 \\ 0 & d & e & 0 & 0 \\ 0 & 0 & f & 0 & 0 \\ \hline 0 & 0 & 0 & a & b \\ 0 & 0 & 0 & 0 & d \end{array}\right]$$

Upper triangular scaling

Whenever $\boldsymbol{A}, P, Q$ scalable *by upper triangulars*, the following terminates:

Algorithm (Triangular scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to P, Q by *upper triangulars*.

- Check if done. Else, scale $\boldsymbol{A}$ from the left by *upper triangular*.
- Check if done. Else, scale $\boldsymbol{A}$ from the right by *upper triangular*.

Upper triangular scaling

Whenever $\boldsymbol{A}, P, Q$ scalable *by upper triangulars*, the following terminates:

Algorithm (Triangular scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to P, Q by *upper triangulars*.

- Check if done. Else, scale $\boldsymbol{A}$ from the left by *upper triangular*.
- Check if done. Else, scale $\boldsymbol{A}$ from the right by *upper triangular*.

Upper triangular scaling

Whenever $\boldsymbol{A}, P, Q$ scalable *by upper triangulars*, the following terminates:

Algorithm (Triangular scaling)

Input: $\boldsymbol{A}, P, Q, \varepsilon$.

Output: ε-scaling of $\boldsymbol{A}$ to P, Q by *upper triangulars*.

- Check if done. Else, scale $\boldsymbol{A}$ from the left by *upper triangular*.
- Check if done. Else, scale $\boldsymbol{A}$ from the right by *upper triangular*.

Running time analysis

Analysis almost identical to (GGOW16, Gu04); progress measure is

$$\operatorname{cap} \tilde{\boldsymbol{A}} = \inf_{X \succ 0} \frac{\det \sum \tilde{A}_i X \tilde{A}_i^\dagger}{\det X}$$

---

1. After the first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \geq C \exp(-10bm)$ where b is total bit-complexity of $\boldsymbol{A}$ (GGOW16 + DM17 + log-convexity of $\operatorname{cap}(\tilde{\boldsymbol{A}})$).
2. If $\boldsymbol{A}$ not ε-scaling,
$$\operatorname{cap} \widetilde{g\boldsymbol{A}} \geq e^{\frac{p_n}{\operatorname{Tr} P}\varepsilon^2} \operatorname{cap} \tilde{\boldsymbol{A}}$$
by AM-GM. Similar for right scaling.
3. Entire time after first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \leq C$.

Running time $O\left(\frac{bm \operatorname{Tr} P}{\varepsilon^2 p_n}\right) = O(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$.

Running time analysis

Analysis almost identical to (GGOW16, Gu04); progress measure is

$$\operatorname{cap} \tilde{\boldsymbol{A}} = \inf_{X \succ 0} \frac{\det \sum \tilde{A}_i X \tilde{A}_i^\dagger}{\det X}$$

---

1. After the first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \geq C \exp(-10bm)$ where b is total bit-complexity of $\boldsymbol{A}$ (GGOW16 + DM17 + log-convexity of $\operatorname{cap}(\tilde{\boldsymbol{A}})$).
2. If $\boldsymbol{A}$ not ε-scaling,

$$\operatorname{cap} \widetilde{g\boldsymbol{A}} \geq e^{\frac{p_n}{\operatorname{Tr} P}\varepsilon^2} \operatorname{cap} \tilde{\boldsymbol{A}}$$

   by AM-GM. Similar for right scaling.
3. Entire time after first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \leq C$.

Running time $O\left(\frac{bm \operatorname{Tr} P}{\varepsilon^2 p_n}\right) = O(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$.

Running time analysis

Analysis almost identical to (GGOW16, Gu04); progress measure is

$$\operatorname{cap} \tilde{\boldsymbol{A}} = \inf_{X \succ 0} \frac{\det \sum \tilde{A}_i X \tilde{A}_i^\dagger}{\det X}$$

---

1. After the first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \geq C \exp(-10bm)$ where b is total bit-complexity of $\boldsymbol{A}$ (GGOW16 + DM17 + log-convexity of $\operatorname{cap}(\tilde{\boldsymbol{A}})$).
2. If $\boldsymbol{A}$ not ε-scaling,

$$\operatorname{cap} \widetilde{g\boldsymbol{A}} \geq e^{\frac{p_n}{\operatorname{Tr} P}\varepsilon^2} \operatorname{cap} \tilde{\boldsymbol{A}}$$

   by AM-GM. Similar for right scaling.
3. Entire time after first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \leq C$.

Running time $O\left(\frac{bm \operatorname{Tr} P}{\varepsilon^2 p_n}\right) = O(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$.

Running time analysis

Analysis almost identical to (GGOW16, Gu04); progress measure is

$$\operatorname{cap} \tilde{\boldsymbol{A}} = \inf_{X \succ 0} \frac{\det \sum \tilde{A}_i X \tilde{A}_i^\dagger}{\det X}$$

---

1. After the first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \geq C \exp(-10bm)$ where b is total bit-complexity of $\boldsymbol{A}$ (GGOW16 + DM17 + log-convexity of $\operatorname{cap}(\tilde{\boldsymbol{A}})$).
2. If $\boldsymbol{A}$ not ε-scaling,

   $$\operatorname{cap} \widetilde{g\boldsymbol{A}} \geq e^{\frac{p_n}{\operatorname{Tr} P}\varepsilon^2} \operatorname{cap} \tilde{\boldsymbol{A}}$$

   by AM-GM. Similar for right scaling.
3. Entire time after first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \leq C$.

Running time $O\left(\frac{bm \operatorname{Tr} P}{\varepsilon^2 p_n}\right) = O(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$.

Running time analysis

Analysis almost identical to (GGOW16, Gu04); progress measure is

$$\operatorname{cap} \tilde{\boldsymbol{A}} = \inf_{X \succ 0} \frac{\det \sum \tilde{A}_i X \tilde{A}_i^\dagger}{\det X}$$

---

1. After the first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \geq C \exp(-10bm)$ where b is total bit-complexity of $\boldsymbol{A}$ (GGOW16 + DM17 + log-convexity of $\operatorname{cap}(\tilde{\boldsymbol{A}})$).
2. If $\boldsymbol{A}$ not ε-scaling,
$$\operatorname{cap} \widetilde{g\boldsymbol{A}} \geq e^{\frac{p_n}{\operatorname{Tr} P}\varepsilon^2} \operatorname{cap} \tilde{\boldsymbol{A}}$$
by AM-GM. Similar for right scaling.
3. Entire time after first step, $\operatorname{cap} \tilde{\boldsymbol{A}} \leq C$.

Running time $O\left(\frac{bm \operatorname{Tr} P}{\varepsilon^2 p_n}\right) = O(b \cdot m \cdot n \cdot \kappa(P)/\varepsilon^2)$.

What about general scalings?

So far:

If the random initial scaling $g\boldsymbol{A}h$ is scalable to P, Q by upper triangulars, the triangular scaling part of Random + Triangular succeeds.

General scalings

It remains to show that the *random* part succeeds:

If $\boldsymbol{A}, P, Q$ scalable by anything, then $g\boldsymbol{A}h, P, Q$ is scalable by upper triangulars with probability at least $2/3$ over choice of g, h.

"Proof:"

Lemma

(g, h) such that $g\boldsymbol{A}h$ is NOT scalable to P, Q by upper triangulars forms an algebraic variety V of degree at most $2 \cdot 2^{2b}$.

If $\boldsymbol{A}$, P, Q scalable, then V must not be everything (the scaling matrices must not lie in V!).

By the Schwartz-Zippel lemma, random (g, h) with entries in $6 \cdot 2^{2b}$ miss V with probability at least $2/3$.

General scalings

It remains to show that the *random* part succeeds:
If $\boldsymbol{A}, P, Q$ scalable by anything, then $g\boldsymbol{A}h, P, Q$ is scalable by upper triangulars with probability at least $2/3$ over choice of g, h.

"Proof:"

Lemma

(g, h) such that $g\boldsymbol{A}h$ is NOT scalable to P, Q by upper triangulars forms an algebraic variety V of degree at most $2 \cdot 2^{2b}$.

If $\boldsymbol{A}$, P, Q scalable, then V must not be everything (the scaling matrices must not lie in V!).
By the Schwartz-Zippel lemma, random (g, h) with entries in $6 \cdot 2^{2b}$ miss V with probability at least $2/3$.

General scalings

It remains to show that the *random* part succeeds:
If $\boldsymbol{A}, P, Q$ scalable by anything, then $g\boldsymbol{A}h, P, Q$ is scalable by upper triangulars with probability at least $2/3$ over choice of g, h.

"Proof:"

Lemma

(g, h) such that $g\boldsymbol{A}h$ is NOT scalable to P, Q by upper triangulars forms an algebraic variety V of degree at most $2 \cdot 2^{2b}$.

If $\boldsymbol{A}$, P, Q scalable, then V must not be everything (the scaling matrices must not lie in V!).
By the Schwartz-Zippel lemma, random (g, h) with entries in $6 \cdot 2^{2b}$ miss V with probability at least $2/3$.

General scalings

It remains to show that the *random* part succeeds:
If $\boldsymbol{A}, P, Q$ scalable by anything, then $g\boldsymbol{A}h, P, Q$ is scalable by upper triangulars with probability at least $2/3$ over choice of g, h.

"Proof:"

Lemma

(g, h) such that $g\boldsymbol{A}h$ is NOT scalable to P, Q by upper triangulars forms an algebraic variety V of degree at most $2 \cdot 2^{2b}$.

If $\boldsymbol{A}$, P, Q scalable, then V must not be everything (the scaling matrices must not lie in V!).

By the Schwartz-Zippel lemma, random (g, h) with entries in $6 \cdot 2^{2b}$ miss V with probability at least $2/3$.

General scalings

It remains to show that the *random* part succeeds:
If $\boldsymbol{A}, P, Q$ scalable by anything, then $g\boldsymbol{A}h, P, Q$ is scalable by upper triangulars with probability at least $2/3$ over choice of g, h.

"Proof:"

Lemma

(g, h) such that $g\boldsymbol{A}h$ is NOT scalable to P, Q by upper triangulars forms an algebraic variety V of degree at most $2 \cdot 2^{2b}$.

If $\boldsymbol{A}$, P, Q scalable, then V must not be everything (the scaling matrices must not lie in V!).
By the Schwartz-Zippel lemma, random (g, h) with entries in $6 \cdot 2^{2b}$ miss V with probability at least $2/3$.

Future work

1. No known poly. time decision algorithms for the operator scaling question (even Brascamp-Lieb) - must scale exponentially close to be sure!
2. Improve ε-dependence to $\log(1/\varepsilon)$ (a way to solve previous two problems).
3. Long term goal: $\text{poly}(\log(1/\varepsilon))$ for one-body quantum marginals - nice consequences:
    - *new* polynomial time null-cone membership algorithms,
    - new membership algorithms for highly complicated polytopes
4. Derandomize.

Future work

1. No known poly. time decision algorithms for the operator scaling question (even Brascamp-Lieb) - must scale exponentially close to be sure!
2. Improve ε-dependence to $\log(1/\varepsilon)$ (a way to solve previous two problems).
3. Long term goal: poly($\log(1/\varepsilon)$) for one-body quantum marginals - nice consequences:
   - *new* polynomial time null-cone membership algorithms,
   - new membership algorithms for highly complicated polytopes
4. Derandomize.

Future work

1. No known poly. time decision algorithms for the operator scaling question (even Brascamp-Lieb) - must scale exponentially close to be sure!
2. Improve ε-dependence to $\log(1/\varepsilon)$ (a way to solve previous two problems).
3. Long term goal: $\mathrm{poly}(\log(1/\varepsilon))$ for one-body quantum marginals - nice consequences:
   - *new* polynomial time null-cone membership algorithms,
   - new membership algorithms for highly complicated polytopes
4. Derandomize.

Future work

1. No known poly. time decision algorithms for the operator scaling question (even Brascamp-Lieb) - must scale exponentially close to be sure!
2. Improve ε-dependence to $\log(1/\varepsilon)$ (a way to solve previous two problems).
3. Long term goal: $\text{poly}(\log(1/\varepsilon))$ for one-body quantum marginals - nice consequences:
    - *new* polynomial time null-cone membership algorithms,
    - new membership algorithms for highly complicated polytopes
4. Derandomize.