Capacity and Scaling Algorithms

Polynomial Capacity: Theory, Applications, Generalizations

Jonathan Leake

Technische Universität Berlin

February 11th, 2021

Outline

1. Matrix scaling
   - Motivation
   - Sinkhorn's scaling algorithm
   - Analysis and connection to capacity
2. Operator scaling
   - Motivation
   - Algorithm for scaling operators
   - Matrix capacity
3. Generalizations and other questions

Outline

The matrix scaling problem

Let M be an $m \times n$ matrix with $\mathbb{R}_+$ entries, and fix $\boldsymbol{r} \in \mathbb{R}_+^m$ and $\boldsymbol{c} \in \mathbb{R}_+^n$.

Definition: A **scaling** of M is given by multiplying M on the left and right by diagonal matrices with positive entries:

$$\text{scaling} = AMB \quad \implies \quad (AMB)_{ij} = a_{ii} m_{ij} b_{jj}.$$

Question: Given M, do there exist such A, B such that the row sums and column sums of AMB are $\boldsymbol{r}$ and $\boldsymbol{c}$ respectively?

Easy: Achieve rows sums by letting $\boldsymbol{\alpha}$ be the row sums of M and apply:

$$A := \operatorname{diag}\left(\frac{r_1}{\alpha_1}, \ldots, \frac{r_m}{\alpha_m}\right) \quad \implies \quad \sum_{j=1}^{n} (AM)_{ij} = \sum_{j=1}^{n} \frac{r_i}{\alpha_i} \cdot m_{ij} = r_i.$$

And same for the columns. **But what about both at the same time?**

Scaling the rows changes the column sums, and vice versa...

Why do we care about matrix scaling?

Application: Deterministic approximation to the permanent. **How?**

Given an $n \times n$ matrix M, set $\boldsymbol{r} = \boldsymbol{c} = \mathbf{1}$. Suppose we have obtained the matrices A, B which scale M to the correct row/column sums.

Since AMB is doubly stochastic, we can use van der Waerden bound:

$$1 \geq \mathrm{per}(AMB) \geq \frac{n!}{n^n} \geq e^{-n} \quad \text{(e.g., recall } \mathrm{Cap}_{\mathbf{1}}(p) \geq p_{\mathbf{1}} \geq \frac{n!}{n^n}\,\mathrm{Cap}_{\mathbf{1}}(p)\text{)}.$$

Now: $\mathrm{per}(AM) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} (AM)_{i,\sigma(i)} = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{ii} m_{i,\sigma(i)} = \det(A)\,\mathrm{per}(M)$.

Similar for B**:** $\mathrm{per}(AMB) = \det(A)\,\mathrm{per}(M)\det(B)$. **Therefore:**

$$[\det(A)\det(B)]^{-1} \geq \mathrm{per}(M) \geq e^{-n}\,[\det(A)\det(B)]^{-1}.$$

This says that $\det(AB)^{-1}$ is an e^n-approximation to the permanent of M.

[Linial-Samorodnitsky-Wigderson '00]: No capacity at the time, but the vdW bound was already proven by Egorychev and Falikman.

How to compute the scaling?

If we have the scaling, then we get an approximation to the permanent.

Questions: How do we compute the A, B? How do we know A, B exist?

Existence: Right off the bat, $\text{per}(M) = 0 \implies$ not scalable. ($\text{per}(M) = 0$ is equivalent to non-existence of perfect matchings in bipartite graph.)

Problem: There exists non-scalable M with $\text{per}(M) > 0$.

Solution: Can *almost* scale when $\text{per}(M) > 0$ [Rothblum-Schneider '89]:

$$A, B \quad \text{such that} \quad \text{row-sums}(AMB) = \boldsymbol{r} \quad \text{and} \quad \text{col-sums}(AMB) = \boldsymbol{c}'$$

with $\|\boldsymbol{c} - \boldsymbol{c}'\| < \epsilon$ for any ϵ.

New problem: For the case of $\boldsymbol{r} = \boldsymbol{c} = \mathbf{1}$ and the permanent, the vdW bound only works for doubly stochastic matrices. **How do we handle "almost doubly stochastic" matrices?** Handle this later...

First: How do we even compute A and B?

Sinkhorn's scaling algorithm

Given M, want to compute A, B so that AMB is almost doubly stochastic.

Sinkhorn's algorithm is a very simple iterative algorithm for M_t:

1. Scale the columns so that col-sums$(M_{t+1}) = \mathbf{1}$.
2. Scale the rows so that row-sums$(M_{t+2}) = \mathbf{1}$ (changes col sums).
3. Repeat iterations until M_t is almost doubly stochastic.

Keep track of $M_t = \cdots A_6 A_4 A_2 M B_1 B_3 B_5 \cdots$, which gives A and B.

Question: How many iterations do we need?

[LSW '00]: If $\mathrm{per}(M) > 0$, then $\mathrm{poly}(n)$ iterations gives M_t with row sums $\mathbf{1}$ and col sums $\boldsymbol{c}_t$ such that $\|\mathbf{1} - \boldsymbol{c}_t\|_2^2$ small (after preprocessing).

Proof idea: When $\|\mathbf{1} - \boldsymbol{c}_t\|_2^2 = C$, iteration scales permanent by $1 + \Omega(C)$. So big C implies big permanent improvement.

Finally: Van der Waerden-type bounds on the permanent for "close" to doubly stochastic give the exponential approximation.

The LSW algorithm

Given M, want to compute A, B so that AMB is almost doubly stochastic.

Main algorithm steps:

1. **Preprocessing:** Scale to get M_1 such that $\text{per}(M_1) \geq \frac{1}{n^n}$.
2. **Sinkhorn:** Apply iterative scaling until $\|\mathbf{1} - \boldsymbol{c}_t\|_2$ is small.
3. **Approximation:** M_t is close to doubly stochastic $\implies \approx e^n$-approx.

Output: $A = A_2 A_4 A_6 \cdots$ and $B = B_1 B_3 B_5 \cdots$ and $\text{per}(M) \approx \det(AB)^{-1}$.

Different "marginals": Similar algorithm given in [LSW '00].

General form of multiplicative iterative scaling algorithms:

1. **Lower bound:** Only need "small" number of steps to get close to DS.
2. **Progress:** Apply Sinkhorn until "marginals" close to DS.
3. **Approximation:** Once close to DS, use vdW-type approximation.

This framework works in more general operator (tensor?) scaling setting.

Analyzing the progress step

Lemma: Given $\boldsymbol{x} \in \mathbb{R}^n_+$ such that $\sum_i x_i = n$ and $\|\mathbf{1} - \boldsymbol{x}\|_2^2 = C$, we have:

$$\prod_{i=1}^n x_i \leq 1 - \frac{C}{2} + O(C^{3/2}) \quad \Longrightarrow \quad \frac{1}{\prod_i x_i} \geq 1 + \Omega(C).$$

Corollary: If M_t has row sums $\boldsymbol{r}_t = \mathbf{1}$ and column sums $\boldsymbol{c}_t$ with $\|\mathbf{1} - \boldsymbol{c}_t\|_2^2 = \epsilon_t$, then $1 \geq \text{per}(M_{t+1}) \geq (1 + \Omega(\epsilon_t)) \cdot \text{per}(M_t)$.

Proof: Note that $\sum_i (\boldsymbol{c}_t)_i = \sum_i (\boldsymbol{r}_t)_i = n$. Scaling columns gives

$$\text{per}(M_{t+1}) = \text{per}\left(M_t \cdot \text{diag}(\boldsymbol{c}_t^{-1})\right) = \text{per}(M_t) \cdot \frac{1}{\prod_i (\boldsymbol{c}_t)_i}.$$

Apply lemma to get $\text{per}(M_{t+1}) \geq (1 + \Omega(\epsilon_t)) \cdot \text{per}(M_t)$.

Now: For $\epsilon_t \geq \frac{1}{n^3}$, apply $O(n^4 \log n)$ steps to get factor of:

$$\left(1 + \Omega\left(\frac{1}{n^3}\right)\right)^{O(n^4 \log n)} \approx e^{O(n \log n)} = O(n^n).$$

Finally: Either ϵ_t becomes small or $O(n^n)$ improvement to permanent.

The LSW algorithm with more detail

Recall the algorithm: (assuming $\text{per}(M) > 0$)

1. **Preprocess** to get $\text{per}(M_1) \geq \frac{1}{n^n} = e^{-\text{poly}(n)}$.
2. **Iterate** $O(n^4 \log n)$ times until $\epsilon_t < \frac{1}{n^3}$ or $O(n^n)$ improvement.
3. If $O(n^n)$ improvement, then $1 \geq \text{per}(M_t) = O(1) \approx 1$.
4. Otherwise $\|\mathbf{1} - \boldsymbol{c}_t\|_2^2 < \frac{1}{n^3} \implies M_t \approx$ doubly stochastic.

Question: What about the last step?

Answer: [LSW '00] gives a vdW-type approximation for close-to-DS M_t.

Generalization: Recall $p(\boldsymbol{x}) := \prod_{i=1}^n \sum_{j=1}^n m_{ij} x_j$ where p is real stable and $p_{\mathbf{1}} = \text{per}(M)$. We have:

- Row sums $= \mathbf{1} \implies p(\mathbf{1}) = \prod_{i=1}^n \sum_{j=1}^n m_{ij} = 1$.
- Column sums $= \boldsymbol{c} \implies \nabla p(\mathbf{1}) = \boldsymbol{c}$.

More general question: Can we bound the coefficient $p_{\mathbf{1}}$ when real stable p is close to being a doubly stochastic polynomial?

Close-to-doubly stochastic real stable polynomials

Theorem (Gurvits-L '20)

Let $p \in \mathbb{R}_+[x_1, \ldots, x_n]$ be a homogeneous polynomial of degree n with $p(\mathbf{1}) = 1$. If p is real stable and $\|\mathbf{1} - \nabla p(\mathbf{1})\|_1 < 2$, then

$$1 \geq \mathrm{Cap}_{\mathbf{1}}(p) = \inf_{\boldsymbol{x}>0} \frac{p(\boldsymbol{x})}{\boldsymbol{x}^{\mathbf{1}}} \geq \left(1 - \frac{\|\mathbf{1} - \nabla p(\mathbf{1})\|_1}{2}\right)^n.$$

Combine with Gurvits' theorem when $\nabla p(\mathbf{1}) = \boldsymbol{c}$:

$$1 \geq \mathrm{Cap}_{\mathbf{1}}(p) \geq p_{\mathbf{1}} \geq \frac{n!}{n^n} \cdot \mathrm{Cap}_{\mathbf{1}}(p) \geq \frac{n!}{n^n} \cdot \left(1 - \frac{\|\mathbf{1} - \boldsymbol{c}\|_1}{2}\right)^n.$$

If $\|\mathbf{1} - \boldsymbol{c}\|_2^2 \leq \frac{1}{n^3}$, then $\|\mathbf{1} - \boldsymbol{c}\|_1 \leq \frac{1}{n}$. **Therefore:**

$$1 \geq p_{\mathbf{1}} \geq \frac{n!}{n^n} \cdot \left(1 - \frac{1}{2n}\right)^n \approx \frac{n!}{n^n} \cdot e^{-\frac{1}{2}} \geq e^{-n}.$$

This gives the final piece of the algorithm for approximating $\mathrm{per}(M)$.

Outline

The operator scaling problem

Let T be a linear operator from $m \times m$ matrices to $n \times n$ matrices which maps PSD matrices to PSD matrices.

Definition: A **scaling** of T is given by PD matrices A, B:

$$\text{scaling} = A^{1/2} T(B^{1/2} X B^{1/2}) A^{1/2}, \quad \text{another PSD-preserving operator.}$$

Question: Given T, do there exist A, B to scale to "doubly stochastic"?

Doubly stochastic operator: $T(I_m) = I_n$ and $T^*(I_n) = I_m$ ($\implies m = n$).

Translated to matrices: $M \cdot \mathbf{1} = \mathbf{1}$ and $M^* \cdot \mathbf{1} = \mathbf{1}$ (doubly stochastic).

As before: Easy to scale one or the other, but what about both? **E.g.:**

$$A := T(I_n)^{-1} \quad \implies \quad \left[A^{1/2} \cdot T \cdot A^{1/2}\right](I_n) = I_n.$$

Scaling via A affects $T^*(I_n)$ and scaling via B affects $T(I_n)$.

Why do we care about operator scaling?

Main operators of study are **completely positive (CP) operators**:

$$T(X) = \sum_{k=1}^{\ell} M_k^* \, X \, M_k \quad \implies \quad T^*(Y) = \sum_{k=1}^{\ell} M_k \, X \, M_k^*,$$

where M_k are any $m \times n$ complex matrices.

Fun fact: Equivalent to $(\mathrm{id}_{k \times k} \otimes T)$ preserving PSD matrices for all k.

First idea [Gurvits '04]: There is an (approximate) scaling if and only if T is **rank non-decreasing**: $\mathrm{rank}(T(X)) \geq \mathrm{rank}(X)$ for all $X \succ 0$.

Matrix case: "Rank non-decreasing" $= \#\{(M\boldsymbol{x})_i = 0\} \geq \#\{x_i = 0\}$ for all $\boldsymbol{x} \in \mathbb{R}_+^n$. This is **Hall marriage condition** $\iff \#\mathrm{pm} = \mathrm{per}(M) > 0$.

I.e.: Rank non-decreasing is operator version of Hall marriage condition.

Summary: Scalability of T is related to some "non-singularity property" of the matrices $M_1, \ldots, M_\ell$.

Why do we care about operator scaling?

Last slide: T is scalable to DS iff $\mathrm{rank}(T(X)) \geq \mathrm{rank}(X)$ for all $X \succ 0$.

CP operator: $$T(X) = \sum_{k=1}^{\ell} M_k^* X M_k \quad \implies \quad T^*(X) = \sum_{k=1}^{\ell} M_k X M_k^*.$$

Why do we care about rank non-decreasing? Equivalent properties (see [Garg-Gurvits-Oliveira-Wigderson '15], Theorem 1.4):

1. $\mathrm{rank}(T(X)) \geq \mathrm{rank}(X)$ for all $X \succ 0$.
2. For some $B_1, \ldots, B_\ell$, the matrix $\sum_{k=1}^{\ell} B_k \otimes M_k$ is non-singular.
3. For some k, the polynomial $\det\left(\sum_{k=1}^{\ell} X_k \otimes M_k\right)$ is not identically 0 where X_k is a $k \times k$ matrix of variables.
4. The "polynomial" $\mathrm{Det}\left(\sum_{k=1}^{\ell} M_k x_k\right)$ is not identically 0, where $x_1, \ldots, x_\ell$ are *non-commuting* variables (non-commutative "Det").
5. The tuple $(M_1, \ldots, M_\ell)$ is not in **null-cone** of left-right action of SL_n^2.

#4: (non-commutative) polynomial identity testing, (NC)PIT: When is the determinant of a matrix of linear forms identically zero?

[Kabanets-Impagliazzo]: Poly-time PIT $\implies$ complexity *lower* bounds.

Gurvits' algorithm

Sinkhorn's algorithm: Alternate scaling rows and columns.

Gurvits' algorithm: Alternate scaling T and T^*:

$$\cdots A_3^{1/2} A_1^{1/2}\, T\left(\cdots B_4^{1/2} B_2^{1/2}\, X\, B_2^{1/2} B_4^{1/2} \cdots\right) A_1^{1/2} A_3^{1/2} \cdots$$

How? Pick $A = T(I_n)^{-1}$ for $\left[A^{1/2}\, T\, A^{1/2}\right](I_n) = I_n$. Pick $B = T^*(I_n)^{-1}$:

$$\begin{aligned}\left[T\left(B^{1/2}\, X\, B^{1/2}\right)\right]^*(I_n) &= \left[\sum_{k=1}^{\ell} M_k^* B^{1/2} X B^{1/2} M_k\right]^*(I_n) \\ &= \left[\sum_{k=1}^{\ell} B^{1/2} M_k X M_k^* B^{1/2}\right](I_n) \\ &= B^{1/2} \cdot T^*(I_n) \cdot B^{1/2} = I_n.\end{aligned}$$

That is: $T(I_n) = I_n$ after odd steps and $T^*(I_n) = I_n$ after even steps.

The general form of the algorithm

Recall the form, for some "measure of progress" μ:

1. **Preprocess:** Scale to T_1 such that $\mu(T_1) \geq e^{-\mathrm{poly}(n)}$.
2. **Iterations:** Iterate $\mathrm{poly}(n)$ times, improving $\mu(T_t)$ multiplicatively by $1 + \frac{1}{O(\mathrm{poly}(n))}$ each time based on "closeness of marginals".
3. **Approximation:** Once "marginals" are close to doubly stochastic, we can approximate. (Approximate what?)

Matrix case: $\mu =$ permanent. Could have also used $\mu = \mathrm{Cap}_{\mathbf{1}}$, since p is doubly stochastic iff $\mathrm{Cap}_{\mathbf{1}}(p) = 1$ and $\mathrm{Cap}_{\mathbf{1}}(p) \leq 1$ otherwise.

Gurvits: Generalize permanent to "quantum permanent" (next slide).

Enough for us: Only need ["marginals" close to doubly stochastic] to imply [we can (almost) scale to doubly stochastic]. **Why?**

Recall: Simply knowing whether that T is (almost) scalable implies $\mathrm{Det}\left(\sum_{k=1}^{\ell} M_k x_k\right) \not\equiv 0$ where the variables are non-commutative (NC-PIT).

Measure of progress: Quantum permanent

Gurvits idea to generalize permanent: "Quantum permanent".

Recall: $\mathrm{per}(M) = \partial_{x_1} \cdots \partial_{x_n}|_{\boldsymbol{x}=0} \prod_{i=1}^n \sum_{j=1}^n m_{ij} x_j$.

Now: $\mathrm{Qper}(T) := \det(\partial_X)|_{X=0} \det(T(X))$ where X is matrix of variables.

Recall: $1 \geq \mathrm{per}(M) \geq \frac{n!}{n^n}$ for doubly stochastic M.

Problem: There is doubly stochastic T such that $\mathrm{Qper}(T) = 0$:

$$T(X) := \frac{1}{2}\left(M_1 X M_1^* + M_2 X M_2^* + M_3 X M_3^*\right)$$

where $M_1 = \begin{bmatrix} 0 & 1 & 0 \\ -1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$, $M_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ -1 & 0 & 0 \end{bmatrix}$, and $M_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & -1 & 0 \end{bmatrix}$. These matrices span the 3×3 skew-symmetric matrices, all of which are singular.

Upshot: Quantum permanent measures PIT, while DS measures NC-PIT.

Related: $\det\left(\begin{bmatrix} 0 & z & w \\ -z & 0 & 1 \\ -w & -1 & 0 \end{bmatrix}\right) \equiv 0$, but $\mathrm{Det}\left(\begin{bmatrix} 0 & z & w \\ -z & 0 & 1 \\ -w & -1 & 0 \end{bmatrix}\right) = zw - wz$.

Matrix capacity

Last slide: Quantum permanent is not a good measure of progress.

What about some kind of capacity? Matrix capacity:

$$\mathrm{Cap}(T) := \inf_{X \succ 0} \frac{\det(T(X))}{\det(X)}.$$

Easy: If $T(I_n) = I_n$, then $\mathrm{Cap}(T) \leq 1$.

[Gurvits '04]: If $T(I_n) = I_n$, then T is doubly stochastic iff $\mathrm{Cap}(T) = 1$.

[Gurvits '04]: The following are equivalent for CP map T.

1. $\mathrm{Cap}(T) > 0$.
2. T is rank non-decreasing.
3. For all $\epsilon > 0$, we have $T_t(I_n) = I_n$ and $\|T_t^*(I_n) - I_n\|_F \leq \epsilon$ for $t \gg 0$.
4. For some t, we have $T_t(I_n) = I_n$ and $\|T_t^*(I_n) - I_n\|_F \leq \frac{1}{n+1}$.

This generalizes the matrix case. So to decide rank-nondecreasing, we just need to scale T to be $\frac{1}{n+1}$-close to doubly stochastic.

Matrix capacity and the scaling algorithm

Matrix case: Polynomial capacity computable via convex programming.

Also: $\mathrm{Cap}_{\mathbf{1}}(p) > 0 \iff \mathrm{per}(M) > 0$ for $p \sim M$ by Gurvits' theorem.

Operator case: Close to doubly stochastic via scaling algorithm.

Then: T is almost scalable iff $\mathrm{Cap}(T) > 0$ iff T rank non-decreasing iff...

Unclear: How to compute capacity directly via convex program?

Analysis of algo [GGOW '15]: Let $T(X) = \sum_{k=1}^{\ell} M_k^* \, X \, M_k$.

1. **"Preprocessing":** If $M_1, \ldots, M_\ell$ have integer entries and $\mathrm{Cap}(T) > 0$, then $\mathrm{Cap}(T) \geq \frac{1}{n^{2n}}$.
2. **Progress:** For $T_t(I_n) = I_n$ and $\|T^*(I_n) - I_n\|_F = \epsilon$, we have $\mathrm{Cap}(T_{t+1}) \geq e^{\Omega(\sqrt{\epsilon})} \cdot \mathrm{Cap}(T_t)$.
3. **Termination:** When $\epsilon \leq \frac{1}{n+1}$, we know that T is (almost) scalable.

For $\epsilon > \frac{1}{n+1}$, we have $\left[e^{\Omega(\frac{1}{\sqrt{n+1}})}\right]^{O(n\sqrt{n}\log n)} = n^{2n} \implies$ poly # iterations.

Crucial: After poly steps, either close to DS or $\mathrm{Cap}(T) = 0$.

Outline

Generalizations and other questions

GGOW algorithm: Used for scaling to doubly stochastic.

Other marginals [Franks '18]: $T(I_n) = P$ and $T^*(I_n) = Q$.

- Generalizes of matrix capacity to $\mathrm{Cap}_A(T)$. When $A = \mathrm{diag}(\boldsymbol{a})$:

$$\text{denominator of } \mathrm{Cap}_A = \prod_{j=1}^{n} \det(X_{[j]})^{a_j - a_{j+1}},$$

  where $X_{[j]}$ is the top-left $j \times j$ submatrix and a_j non-increasing.
- Seems different than continuous capacity. **Connection?**
- Seems related to the Gelfand-Tsetlin polytope. **Connection?**

Tensor scaling [Bürgisser-Franks-Garg-Oliveira-Walter-Wigderson]:
Given $\phi \in V^{\otimes m}$, act on each tensor component iteratively in succession:

$$\sum_i (\boldsymbol{v}_i \otimes \boldsymbol{w}_i \otimes \cdots) \to \sum_i (A_1 \boldsymbol{v}_i \otimes \boldsymbol{w}_i \otimes \cdots) \to \sum_i (A_1 \boldsymbol{v}_i \otimes A_2 \boldsymbol{w}_i \otimes \cdots) \to \cdots$$

Invariant theory connections: Next week or the week after.