Partition functions and Scaling

Alex Samorodnitsky

July 18, 2018

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

- The partition function of $\mathcal{F} \subseteq 2^{[N]}$ is

$$p_{\mathcal{F}}(x_1, \ldots, x_N) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$$

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

- The partition function of $\mathcal{F} \subseteq 2^{[N]}$ is

$$p_{\mathcal{F}}(x_1, \ldots, x_N) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$$

- We want to compute $p_{\mathcal{F}}$.

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

- The partition function of $\mathcal{F} \subseteq 2^{[N]}$ is

$$p_{\mathcal{F}}(x_1, \ldots, x_N) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$$

- We want to compute (or at least approximate...) $p_{\mathcal{F}}$.

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

- The partition function of $\mathcal{F} \subseteq 2^{[N]}$ is

$$p_{\mathcal{F}}(x_1, \ldots, x_N) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$$

- We want to compute (or at least approximate...) $p_{\mathcal{F}}$.
- An easy example: $\mathcal{F} = 2^{[N]}$.

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

- The partition function of $\mathcal{F} \subseteq 2^{[N]}$ is

$$p_{\mathcal{F}}(x_1, ..., x_N) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$$

- We want to compute (or at least approximate...) $p_{\mathcal{F}}$.
- An easy example: $\mathcal{F} = 2^{[N]}$. Here $p_{\mathcal{F}} = \prod_{i=1}^{N} (1 + x_i)$.

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

- The partition function of $\mathcal{F} \subseteq 2^{[N]}$ is

$$p_{\mathcal{F}}(x_1, \ldots, x_N) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$$

- We want to compute (or at least approximate...) $p_{\mathcal{F}}$.
- An easy example: $\mathcal{F} = 2^{[N]}$. Here $p_{\mathcal{F}} = \prod_{i=1}^{N} (1 + x_i)$.
- A more complicated case: the permanent.

Computing partition functions

"Combinatorics and Complexity of Partition functions", B'2016

- The partition function of $\mathcal{F} \subseteq 2^{[N]}$ is

$$p_{\mathcal{F}}(x_1, \ldots, x_N) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$$

- We want to compute (or at least approximate...) $p_{\mathcal{F}}$.
- An easy example: $\mathcal{F} = 2^{[N]}$. Here $p_{\mathcal{F}} = \prod_{i=1}^{N}(1 + x_i)$.
- A more complicated case: the permanent. This will be our main example.

- Permanents.

- Permanents.
- Contingency tables and integer flows.

- Permanents.
- Contingency tables and integer flows.
- Mixed discriminants.

Permanent: Definition

- The permanent of an $n \times n$ matrix $A = (a_{ij})$ is

$$Per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}$$

Permanent: Definition

- The permanent of an $n \times n$ matrix $A = (a_{ij})$ is

$$Per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}$$

- Like the determinant, but without the signs.

Permanent: Definition

- The permanent of an $n \times n$ matrix $A = (a_{ij})$ is

$$Per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}$$

- Like the determinant, but without the signs.
- Permanent is a partition function in variables a_{ij}, $1 \leq i, j \leq n$:

Permanent: Definition

- The permanent of an $n \times n$ matrix $A = (a_{ij})$ is

$$Per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}$$

- Like the determinant, but without the signs.
- Permanent is a partition function in variables a_{ij}, $1 \leq i, j \leq n$: $N = n^2$, $\mathcal{F} = S_n$.

Permanent: Definition

- The permanent of an $n \times n$ matrix $A = (a_{ij})$ is

$$Per(A) = \sum_{\sigma \in S_n} \prod_{i=1}^{n} a_{i\sigma(i)}$$

- Like the determinant, but without the signs.
- Permanent is a partition function in variables a_{ij}, $1 \leq i, j \leq n$:

$N = n^2$, $\mathcal{F} = S_n$, $\sigma \leftrightarrow \{(i, \sigma(i)) : 1 \leq i \leq n\} \subseteq [n]^2$.

Permanent - examples

$$Per\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} = Det\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} = 1$$

Permanent - examples

$$Per \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} = Det \begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} = 1$$

$$Per \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix} = n! \neq Det \begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$$

Permanent - examples

$$Per\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} = Det\begin{pmatrix} 1 & 0 & \dots & 0 \\ 0 & 1 & \dots & 0 \\ \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 1 \end{pmatrix} = 1$$

$$Per\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix} = n! \neq Det\begin{pmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \vdots & \vdots \\ 1 & 1 & \dots & 1 \end{pmatrix}$$

- Examples are not easy to come by...

Permanents and counting

- If A is the adjacency matrix of a bipartite graph, $per(A)$ is the number of perfect matchings in the graph.

Permanents and counting

- If A is the adjacency matrix of a bipartite graph, $per(A)$ is the number of perfect matchings in the graph.
- Applications in statistical physics if the graph is a grid.

Permanents and counting

- If A is the adjacency matrix of a bipartite graph, $per(A)$ is the number of perfect matchings in the graph.
- Applications in statistical physics if the graph is a grid.
- In fact, if we can compute permanents, we can compute 'anything'.

Permanents and counting

- If A is the adjacency matrix of a bipartite graph, $per(A)$ is the number of perfect matchings in the graph.
- Applications in statistical physics if the graph is a grid.
- In fact, if we can compute permanents, we can compute 'anything'.
- So we would like to compute permanents.

How to compute the permanent?

- Via the definition: $n!$

How to compute the permanent?

- Via the definition: $n!$
- Inclusion-Exclusion, aka Ryser's formula: 2^n

How to compute the permanent?

- Via the definition: $n!$
- Inclusion-Exclusion, aka Ryser's formula: 2^n
- Faster computation? Polya: try to use similarity to determinant.

How to compute the permanent?

- Via the definition: $n!$
- Inclusion-Exclusion, aka Ryser's formula: 2^n
- Faster computation? Polya: try to use similarity to determinant.
- Polya: can't do this by adding signs to the entries.

How to compute the permanent?

- Via the definition: $n!$
- Inclusion-Exclusion, aka Ryser's formula: 2^n
- Faster computation? Polya: try to use similarity to determinant.
- Polya: can't do this by adding signs to the entries. MM'61: Any linear forms, MR'04: not even with quadratic blow-up.

Permanent is hard to compute

- Valiant'79: The permanent is # P-hard to compute.

Permanent is hard to compute

- Valiant'79: The permanent is # P-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.

Permanent is hard to compute

- Valiant'79: The permanent is # P-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.
- This means that permanent is very good at counting things.

Permanent is hard to compute

- Valiant'79: The permanent is $\#P$-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.
- This means that permanent is very good at counting things.
- ...Can we approximate the permanent?

Permanent is hard to compute

- Valiant'79: The permanent is # P-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.
- This means that permanent is very good at counting things.
- ...Can we approximate the permanent? Yes, we can if the matrix is nonnegative JSV'01.

Permanent is hard to compute

- Valiant'79: The permanent is $\# P$-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.
- ...Can we approximate the permanent? Yes, we can if the matrix is nonnegative JSV'01.
- What about more general matrices?

Permanent is hard to compute

- Valiant'79: The permanent is $\# P$-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.
- ...Can we approximate the permanent? Yes, we can if the matrix is nonnegative JSV'01.
- Computing / approximating permanents of general real-valued matrices seems hopeless.

Permanent is hard to compute

- Valiant'79: The permanent is # P-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.
- ...Can we approximate the permanent? Yes, we can if the matrix is nonnegative JSV'01.
- Computing / approximating permanents of general real-valued matrices seems hopeless.
- It would be interesting though to quantify this feeling.

Permanent is hard to compute

- Valiant'79: The permanent is # P-hard to compute.
- Even for $0-1$ matrices with only 3 ones in a row.
- ...Can we approximate the permanent? Yes, we can if the matrix is nonnegative JSV'01.
- Computing / approximating permanents of general real-valued matrices seems hopeless.
- It would be interesting though to quantify this feeling. Conjecture AA'10: Hard to approximate the permanent of a random Gaussian matrix.

How to approximate the permanent?

- Given a nonnegative matrix A efficiently compute $F(A)$ so that

How to approximate the permanent?

- Given a nonnegative matrix A efficiently compute $F(A)$ so that

$$F(A) \quad \leq \quad Per(A) \quad \leq \quad C \cdot F(A)$$

How to approximate the permanent?

- Given a nonnegative matrix A efficiently compute $F(A)$ so that

$$F(A) \quad \leq \quad Per(A) \quad \leq \quad C \cdot F(A)$$

- There is a randomized algorithm resolving the problem for any C due to JSV'01.

How to approximate the permanent?

- Given a nonnegative matrix A efficiently compute $F(A)$ so that

$$F(A) \quad \leq \quad Per(A) \quad \leq \quad C \cdot F(A)$$

- There is a randomized algorithm resolving the problem for any C due to JSV'01.
- The best deterministic approximation of $C \approx 2^n$ is obtained in LSW'98, GS'14 via matrix scaling.

Some ancient history

- The year is 1997.

Some ancient history

- The year is 1997. The problem of permanent approximation has not been resolved yet. There are two main approaches, both probabilistic:

Some ancient history

- The year is 1997. The problem of permanent approximation has not been resolved yet. There are two main approaches, both probabilistic:
- MCMC:

Some ancient history

- The year is 1997. The problem of permanent approximation has not been resolved yet. There are two main approaches, both probabilistic:
- MCMC:
  - Broder'86: A random walk on perfect matchings of a bipartite graph, converging rapidly to uniform distribution will approximate the number of matchings.

Some ancient history

- The year is 1997. The problem of permanent approximation has not been resolved yet. There are two main approaches, both probabilistic:
- MCMC:
  - Broder'86: A random walk on perfect matchings of a bipartite graph, converging rapidly to uniform distribution will approximate the number of matchings.
  - JS'89: A better random walk. Tools for proving rapid convergence, rapid convergence for some graphs.

Some ancient history

- The year is 1997. The problem of permanent approximation has not been resolved yet. There are two main approaches, both probabilistic:
- MCMC:
  - Broder'86: A random walk on perfect matchings of a bipartite graph, converging rapidly to uniform distribution will approximate the number of matchings.
  - JS'89: A better random walk. Tools for proving rapid convergence, rapid convergence for some graphs.
  - ... (In the future) JSV'01 - A more sophisticated random walk + analysis resolving the question completely.

Some ancient history

- The year is 1997. The problem of permanent approximation has not been resolved yet. There are two main approaches, both probabilistic:
- MCMC:
  - Broder'86: A random walk on perfect matchings of a bipartite graph, converging rapidly to uniform distribution will approximate the number of matchings.
  - JS'89: A better random walk. Tools for proving rapid convergence, rapid convergence for some graphs.
  - ... (In the future) JSV'01 - A more sophisticated random walk + analysis resolving the question completely.
- And an approach exploiting the similarity of permanent and determinant.

Approximating the permanent via determinants

Old history

- GG'78: To estimate the permanent of $A = (a_{ij})$, let

$$B = \left(\epsilon_{ij} \cdot \sqrt{a_{ij}}\right)$$

ϵ_{ij} are independent random variables, with mean zero and variance 1.

Approximating the permanent via determinants

Old history

- GG'78: To estimate the permanent of $A = \left(a_{ij}\right)$, let

$$B = \left(\epsilon_{ij} \cdot \sqrt{a_{ij}}\right)$$

ϵ_{ij} are independent random variables, with mean zero and variance 1.

- Then

$$Per(A) = \mathbb{E}\ Det^2(B)$$

Approximating the permanent via determinants

Old history

- GG'78: To estimate the permanent of $A = (a_{ij})$, let

$$B = (\epsilon_{ij} \cdot \sqrt{a_{ij}})$$

ϵ_{ij} are independent random variables, with mean zero and variance 1.

- Then

$$Per(A) = \mathbb{E}\, Det^2(B)$$

- It only remains to show $Det^2(B)$ is well-concentrated.

Approximating the permanent via determinants

Old history

- GG'78: To estimate the permanent of $A = (a_{ij})$, let

$$B = (\epsilon_{ij} \cdot \sqrt{a_{ij}})$$

ϵ_{ij} are independent random variables, with mean zero and variance 1.

- Then

$$Per(A) = \mathbb{E}\, Det^2(B)$$

- It "only" remains to show $Det^2(B)$ is well-concentrated.

Approximating the permanent via determinants

Old history

- GG'78: To estimate the permanent of $A = (a_{ij})$, let

$$B = (\epsilon_{ij} \cdot \sqrt{a_{ij}})$$

ϵ_{ij} are independent random variables, with mean zero and variance 1.

- Then

$$Per(A) = \mathbb{E}\, Det^2(B)$$

- It "only" remains to show $Det^2(B)$ is well-concentrated.
- GG'78, KKLLL'93: take ϵ_{ij} to be random roots of unity. Not very good.

Approximating the permanent via determinants

Old history

- GG'78: To estimate the permanent of $A = (a_{ij})$, let

$$B = \left(\epsilon_{ij} \cdot \sqrt{a_{ij}}\right)$$

ϵ_{ij} are independent random variables, with mean zero and variance 1.

- Then

$$Per(A) = \mathbb{E}\, Det^2(B)$$

- It "only" remains to show $Det^2(B)$ is well-concentrated.
- GG'78, KKLLL'93: take ϵ_{ij} to be random roots of unity. Not very good.
- B'96: take ϵ_{ij} to be Gaussian. An exponential approximation.

Approximating the permanent via determinants

Old history

- Then

$$Per(A) = \mathbb{E}\, Det^2(B)$$

- It "only" remains to show $Det^2(B)$ is well-concentrated.
- GG'78, KKLLL'93: take ϵ_{ij} to be random roots of unity. Not very good.
- B'96: take ϵ_{ij} to be Gaussian. An exponential approximation.
- That is, a value $F(A)$ such that w.h.p.

$$F(A) \leq Per(A) \leq c^n \cdot F(A) \quad c \approx 4$$

Approximating the permanent via determinants

Old history

- Then

$$Per(A) = \mathbb{E}\, Det^2(B)$$

- GG'78, KKLLL'93: take ϵ_{ij} to be random roots of unity. Not very good.
- B'96: take ϵ_{ij} to be Gaussian. An exponential approximation.
- That is, a value $F(A)$ such that w.h.p.

$$F(A) \leq Per(A) \leq c^n \cdot F(A) \quad c \approx 4$$

The important thing about 4 is that it is bigger than e.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- If A is doubly stochastic,

$$Per(A) \leq 1$$

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- If A is doubly stochastic,

$$Per(A) \leq 1$$

- and

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- If A is doubly stochastic,

$$Per(A) \leq 1$$

- and

Egorychev, Falikman'81:

$$Per(A) \geq \frac{n!}{n^n} > e^{-n}$$

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case:

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}. This gives e^{n}-approximation.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}. This gives e^{n}-approximation.
- This has been the starting point of LSW '98.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}. This gives e^{n}-approximation.
- This has been the starting point of LSW '98. There the general case is reduced to the doubly stochastic case, obtaining deterministic e^{n}-approximation for the permanent of a nonnegative matrix.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}. This gives e^{n}-approximation.
- This has been the starting point of LSW '98. There the general case is reduced to the doubly stochastic case, obtaining deterministic e^{n}-approximation for the permanent of a nonnegative matrix. This reduction is achieved by matrix scaling.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}. This gives e^n-approximation.
- This has been the starting point of LSW '98. There the general case is reduced to the doubly stochastic case, obtaining deterministic e^n-approximation for the permanent of a nonnegative matrix. This reduction is achieved by matrix scaling.
- Back to determinants.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}. This gives e^{n}-approximation.
- This has been the starting point of LSW '98. There the general case is reduced to the doubly stochastic case, obtaining deterministic e^{n}-approximation for the permanent of a nonnegative matrix. This reduction is achieved by matrix scaling.
- Back to determinants. B'99: take ϵ_{ij} to be quaternion Gaussian.

A simple case - doubly stochastic matrices

- A doubly stochastic matrix is a nonnegative matrix with row and columns summing to 1.
- $e^{-n} < Per(A) \leq 1$.
- A simple deterministic algorithm in this case: Always return e^{-n}. This gives e^n-approximation.
- This has been the starting point of LSW '98. There the general case is reduced to the doubly stochastic case, obtaining deterministic e^n-approximation for the permanent of a nonnegative matrix. This reduction is achieved by matrix scaling.
- Back to determinants. B'99: take ϵ_{ij} to be quaternion Gaussian. A 1.2^n-approximation.

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:
  - Input: A matrix A with positive entries.

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:
  - Input: A matrix A with positive entries.
  - Odd iteration: Normalize (scale) all rows to be stochastic.

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:
  - Input: A matrix A with positive entries.
  - Odd iteration: Normalize (scale) all rows to be stochastic.
  - Even iteration: Normalize (scale) all columns to be stochastic.

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:
  - Input: A matrix A with positive entries.
  - Odd iteration: Normalize (scale) all rows to be stochastic.
  - Even iteration: Normalize (scale) all columns to be stochastic.
- Get a sequence of matrices: $A_0 = A$, $A_1, ...A_t, ...$.

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:
  - Input: A matrix A with positive entries.
  - Odd iteration: Normalize (scale) all rows to be stochastic.
  - Even iteration: Normalize (scale) all columns to be stochastic.
- Get a sequence of matrices: $A_0 = A$, $A_1, \ldots A_t, \ldots$.
- Questions: Does it converge to a doubly stochastic matrix?

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:
  - Input: A matrix A with positive entries.
  - Odd iteration: Normalize (scale) all rows to be stochastic.
  - Even iteration: Normalize (scale) all columns to be stochastic.
- Get a sequence of matrices: $A_0 = A$, $A_1, ...A_t, ...$.
- Questions: Does it converge to a doubly stochastic matrix? If yes, how fast?

A simple iterative algorithm to make a matrix doubly-stochastic

- The algorithm:
  - Input: A matrix A with positive entries.
  - Odd iteration: Normalize (scale) all rows to be stochastic.
  - Even iteration: Normalize (scale) all columns to be stochastic.
- Get a sequence of matrices: $A_0 = A$, $A_1, \ldots A_t, \ldots$.
- Questions: Does it converge to a doubly stochastic matrix? If yes, how fast? Can we keep track of the permanent?

Sinkorn's scaling

- The answers to all the questions above are positive (when suitably qualified).

Sinkorn’s scaling

- The answers to all the questions above are positive (when suitably qualified). Some of them were given by S’64 who was the first to propose and analyze this algorithm.

Sinkorn's scaling

- The answers to all the questions above are positive (when suitably qualified). Some of them were given by S'64 who was the first to propose and analyze this algorithm.
- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even.

Sinkorn's scaling

- The answers to all the questions above are positive (when suitably qualified). Some of them were given by S'64 who was the first to propose and analyze this algorithm.
- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per\,(A_{k+1}) \;=\; \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per\,(A_k)$$

Sinkorn's scaling

- The answers to all the questions above are positive (when suitably qualified). Some of them were given by S'64 who was the first to propose and analyze this algorithm.
- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per\,(A_{k+1}) \;=\; \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per\,(A_k)$$

Sinkorn's scaling

- The answers to all the questions above are positive (when suitably qualified). Some of them were given by S'64 who was the first to propose and analyze this algorithm.
- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per(A_{k+1}) = \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per(A_k)$$

- Hence the permanent is trackable.

Sinkorn's scaling

- The answers to all the questions above are positive (when suitably qualified). Some of them were given by S'64 who was the first to propose and analyze this algorithm.
- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per\,(A_{k+1}) \;=\; \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per\,(A_k)$$

- Hence the permanent is trackable. In fact, for every i, j, k:

$$A_k(i,j) \;=\; \lambda_i^{(k)} A(i,j) \mu_j^{(k)}$$

Sinkorn's scaling

- The answers to all the questions above are positive (when suitably qualified). Some of them were given by S'64 who was the first to propose and analyze this algorithm.
- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per\,(A_{k+1}) \;=\; \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per\,(A_k)$$

- Hence the permanent is trackable. In fact, for every i, j, k:

$$A_k(i,j) \;=\; \lambda_i^{(k)} A(i,j) \mu_j^{(k)} \;\Rightarrow\; Per\,(A_k) \;=\; \prod_i \lambda_i^{(k)} \cdot \prod_j \mu_j^{(k)} \cdot Per(A)$$

Sinkorn's scaling

- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per\left(A_{k+1}\right) \;=\; \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per\left(A_k\right) \;>\; Per\left(A_k\right)$$

Sinkorn's scaling

- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per\,(A_{k+1}) \;=\; \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per\,(A_k) \;>\; Per\,(A_k)$$

- It is also increasing, which indicates that the sequence $\{A_k\}_k$ converges to a doubly stochastic matrix.

Sinkorn's scaling

- Let $r_1, ..., r_n$ be the row sums of A_k, $k > 1$ even. Then

$$Per\,(A_{k+1}) \;=\; \frac{1}{\prod_{i=1}^{n} r_i} \cdot Per\,(A_k) \;>\; Per\,(A_k)$$

- It is also increasing, which indicates that the sequence $\{A_k\}_k$ converges to a doubly stochastic matrix. The rate of convergence can be slow, but this could be handled with some preprocessing.

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently. (More about this later.)

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.
- Corollary: The permanent of a positive matrix can be efficiently approximated within a factor of e^n.

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

- Corollary: The permanent of a positive matrix can be efficiently approximated within a factor of e^n.

Proof:

1. Given A, find the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$.

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

- Corollary: The permanent of a positive matrix can be efficiently approximated within a factor of e^n.

Proof:

1. Given A, find the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$. $B = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic.

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

- Corollary: The permanent of a positive matrix can be efficiently approximated within a factor of e^n.

Proof:

1. Given A, find the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$. $B = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic. Hence

$$e^{-n} < Per(B) \leq 1$$

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

- Corollary: The permanent of a positive matrix can be efficiently approximated within a factor of e^n.

Proof:

1. Given A, find the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$. $B = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic. Hence

$$e^{-n} < Per(B) = \prod_i \lambda_i \cdot \prod_j \mu_j \cdot Per(A) \leq 1$$

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

- Corollary: The permanent of a positive matrix can be efficiently approximated within a factor of e^n.

Proof:

1. Given A, find the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$. $B = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic. Hence

$$e^{-n} \ < \ Per(B) \ = \prod_i \lambda_i \cdot \prod_j \mu_j \cdot Per(A) \ \leq \ 1$$

2. Return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.

Matrix scaling

- Let $A = (a_{ij})$ be an $n \times n$ matrix with positive entries. Then
  - There exist 'unique' positive numbers $\lambda_1, ..., \lambda_n$ and $\mu_1, ..., \mu_n$ such that the matrix $(\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

- Corollary: The permanent of a positive matrix can be efficiently approximated within a factor of e^n.

Proof:

1. Given A, find the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$. $B = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic. Hence

$$e^{-n} < Per(B) = \prod_i \lambda_i \cdot \prod_j \mu_j \cdot Per(A) \leq 1$$

2. Return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$. Done.

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, ..., x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, ..., x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$:
- Find a transformation (scaling) of the underlying object

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, ..., x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$:
- Find a transformation (scaling) of the underlying object, that is a change of variables $y_i = \phi_i(x_1, ..., x_n)$

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, ..., x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$:
- Find a transformation (scaling) of the underlying object, that is a change of variables $y_i = \phi_i(x_1, ..., x_n)$, so that
  - The functions $y_i = \phi_i(x_1, ..., x_n)$ are efficiently computable.

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, ..., x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$:
- Find a transformation (scaling) of the underlying object, that is a change of variables $y_i = \phi_i(x_1, ..., x_n)$, so that
  - The functions $y_i = \phi_i(x_1, ..., x_n)$ are efficiently computable.
  - $p_{\mathcal{F}}(y_1, ..., y_n) = S(x_1, ..., x_n) \cdot p_{\mathcal{F}}(x_1, ..., x_n)$, where the scaling factor $S(x_1, ..., x_n)$ is efficiently computable.

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, \ldots, x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$:
- Find a transformation (scaling) of the underlying object, that is a change of variables $y_i = \phi_i(x_1, \ldots, x_n)$, so that
  - The functions $y_i = \phi_i(x_1, \ldots, x_n)$ are efficiently computable.
  - $p_{\mathcal{F}}(y_1, \ldots, y_n) = S(x_1, \ldots, x_n) \cdot p_{\mathcal{F}}(x_1, \ldots, x_n)$, where the scaling factor $S(x_1, \ldots, x_n)$ is efficiently computable.
  - The function $p_{\mathcal{F}}(y_1, \ldots, y_n)$ is well-behaved

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, ..., x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$:
- Find a transformation (scaling) of the underlying object, that is a change of variables $y_i = \phi_i(x_1, ..., x_n)$, so that
  - The functions $y_i = \phi_i(x_1, ..., x_n)$ are efficiently computable.
  - $p_{\mathcal{F}}(y_1, ..., y_n) = S(x_1, ..., x_n) \cdot p_{\mathcal{F}}(x_1, ..., x_n)$, where the scaling factor $S(x_1, ..., x_n)$ is efficiently computable.
  - The function $p_{\mathcal{F}}(y_1, ..., y_n)$ is well-behaved (well-concentrated) on its domain.

Scaling as an approach

Abstracting out

- Given a partition function $p_{\mathcal{F}}(x_1, ..., x_n) = \sum_{S \in \mathcal{F}} \prod_{i \in S} x_i$:
- Find a transformation (scaling) of the underlying object, that is a change of variables $y_i = \phi_i(x_1, ..., x_n)$, so that
  - The functions $y_i = \phi_i(x_1, ..., x_n)$ are efficiently computable.
  - $p_{\mathcal{F}}(y_1, ..., y_n) = S(x_1, ..., x_n) \cdot p_{\mathcal{F}}(x_1, ..., x_n)$, where the scaling factor $S(x_1, ..., x_n)$ is efficiently computable.
  - The function $p_{\mathcal{F}}(y_1, ..., y_n)$ is well-behaved (well-concentrated) on its domain. This would usually mean that the transformed object has some regularity properties.

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$.

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$.

This gives a multiplicative range of e^n, which is not so good.

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?
- Consider the bounds we use.

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?
- Consider the bounds we use. The lower bound of e^{-n} is essentially tight for the matrix $J = \left(\frac{1}{n}\right)$

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?
- Consider the bounds we use. The lower bound of e^{-n} is essentially tight for the matrix $J = \left(\frac{1}{n}\right)$ and the upper bound of 1 is tight for the identity matrix I.

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?
- Consider the bounds we use. The lower bound of e^{-n} is essentially tight for the matrix $J = \left(\frac{1}{n}\right)$ and the upper bound of 1 is tight for the identity matrix I.
- Most matrices are far from both I and J.

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?
- Consider the bounds we use. The lower bound of e^{-n} is essentially tight for the matrix $J = \left(\frac{1}{n}\right)$ and the upper bound of 1 is tight for the identity matrix I.
- Most matrices are far from both I and J. (In fact, rather closer to J than to I).

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?
- Consider the bounds we use. The lower bound of e^{-n} is essentially tight for the matrix $J = \left(\frac{1}{n}\right)$ and the upper bound of 1 is tight for the identity matrix I.
- Most matrices are far from both I and J.
- Can we "look" at a doubly stochastic matrix and estimate its permanent better?

Concentration of the permanent - can we improve it?

- A key fact: if A is doubly stochastic, then $e^{-n} < Per(A) \leq 1$. This gives a multiplicative range of e^n, which is not so good. Can we improve it?
- Consider the bounds we use. The lower bound of e^{-n} is essentially tight for the matrix $J = \left(\frac{1}{n}\right)$ and the upper bound of 1 is tight for the identity matrix I.
- Most matrices are far from both I and J.
- Can we "look" at a doubly stochastic matrix and estimate its permanent better? Need "non-blackbox bounds" for the permanent.

A small improvement

- A step in this direction was taken in GS'14. Let $A = (a_{ij})$ be doubly stochastic.

A small improvement

- A step in this direction was taken in GS'14. Let $A = (a_{ij})$ be doubly stochastic.
- A lower bound:

$$Per(A) \geq \prod_{i,j=1}^{n} \left(1 - a_{ij}\right)^{1-a_{ij}}$$

A small improvement

- A step in this direction was taken in GS'14. Let $A = (a_{ij})$ be doubly stochastic.
- A lower bound:

$$Per(A) \geq \prod_{i,j=1}^{n} (1 - a_{ij})^{1-a_{ij}}$$

- An upper bound: Let $a_1, ..., a_n$ be the rows of A.

A small improvement

- A step in this direction was taken in GS'14. Let $A = (a_{ij})$ be doubly stochastic.
- A lower bound:

$$Per(A) \geq \prod_{i,j=1}^{n} (1 - a_{ij})^{1-a_{ij}}$$

- An upper bound: Let $a_1, ..., a_n$ be the rows of A. For a suitable chosen convex function ψ on $[0, 1]$ defining an Orlicz norm $\| \cdot \|_\psi$

A small improvement

- A step in this direction was taken in GS'14. Let $A = (a_{ij})$ be doubly stochastic.
- A lower bound:

$$Per(A) \;\geq\; \prod_{i,j=1}^{n} (1 - a_{ij})^{1-a_{ij}}$$

- An upper bound: Let $a_1, ..., a_n$ be the rows of A. For a suitable chosen convex function ψ on $[0, 1]$ defining an Orlicz norm $\|\cdot\|_\psi$ (a generalization of ℓ_p norms)

A small improvement

- A step in this direction was taken in GS'14. Let $A = (a_{ij})$ be doubly stochastic.
- A lower bound:

$$Per(A) \geq \prod_{i,j=1}^{n} (1 - a_{ij})^{1-a_{ij}}$$

- An upper bound: Let $a_1, ..., a_n$ be the rows of A. For a suitable chosen convex function ψ on $[0, 1]$ defining an Orlicz norm $\| \cdot \|_\psi$ (a generalization of ℓ_p norms) holds

$$Per(A) \leq \prod_{i=1}^{n} \|a_i\|_\psi$$

A small improvement

We have

$$\prod_{i,j=1}^{n} \left(1 - a_{ij}\right)^{1-a_{ij}} \leq Per(A) \leq \prod_{i=1}^{n} \|a_i\|_{\psi}$$

A small improvement

We have

$$\prod_{i,j=1}^{n} \left(1 - a_{ij}\right)^{1-a_{ij}} \leq Per(A) \leq \prod_{i=1}^{n} \|a_i\|_\psi$$

- The ratio between the bounds is about 2^n, giving an 2^n-approximation for the permanent.

A small improvement

We have

$$\prod_{i,j=1}^{n} \left(1 - a_{ij}\right)^{1-a_{ij}} \leq Per(A) \leq \prod_{i=1}^{n} \|a_i\|_\psi$$

- The ratio between the bounds is about 2^n, giving an 2^n-approximation for the permanent.
- The upper bound we state is far from being optimal.

A small improvement

We have

$$\prod_{i,j=1}^{n} \left(1 - a_{ij}\right)^{1-a_{ij}} \leq Per(A) \leq \prod_{i=1}^{n} \|a_i\|_{\psi}$$

- The ratio between the bounds is about 2^n, giving an 2^n-approximation for the permanent.
- The upper bound we state is far from being optimal. We next describe a famous upper bound for $0 - 1$ matrices and some of its applications.

Bregman's upper bound

- B'73: Let A be a $0-1$ matrix with row sums r_i

Bregman's upper bound

- B'73: Let A be a $0-1$ matrix with row sums r_i then

$$Per(A) \leq \prod_{i=1}^{n} (r_i!)^{1/r_i}$$

Bregman's upper bound

- B'73: Let A be a $0-1$ matrix with row sums r_i then

$$Per(A) \;\leq\; \prod_{i=1}^{n} (r_i!)^{1/r_i}$$

- This is tight for a block-diagonal matrix with blocks of size $r_i \times r_i$.

Bregman's upper bound

- B'73: Let A be a $0-1$ matrix with row sums r_i then

$$Per(A) \leq \prod_{i=1}^{n} (r_i!)^{1/r_i}$$

- Corollary. If A is a row-stochastic matrix with maximal i-th row element $m_i \leq \frac{1}{r_i}$, where $r_i \in \mathbb{N}$, then

Bregman's upper bound

- B'73: Let A be a $0-1$ matrix with row sums r_i then

$$Per(A) \;\leq\; \prod_{i=1}^{n}(r_i!)^{1/r_i}$$

- Corollary. If A is a row-stochastic matrix with maximal i-th row element $m_i \leq \frac{1}{r_i}$, where $r_i \in \mathbb{N}$, then

$$Per(A) \;\leq\; \left(\prod_{i=1}^{n} r_i\right)^{-1} \cdot \prod_{i=1}^{n}(r_i!)^{1/r_i}$$

Bregman's upper bound

- B'73: Let A be a $0-1$ matrix with row sums r_i then

$$Per(A) \;\leq\; \prod_{i=1}^{n} (r_i!)^{1/r_i}$$

- Corollary. If A is a row-stochastic matrix with maximal i-th row element $m_i \leq \frac{1}{r_i}$, where $r_i \in \mathbb{N}$, then

$$Per(A) \;\leq\; \left(\prod_{i=1}^{n} r_i\right)^{-1} \cdot \prod_{i=1}^{n} (r_i!)^{1/r_i}$$

- This is easy to see by a simple variation argument.

Bregman's upper bound

- B'73: Let A be a $0-1$ matrix with row sums r_i then

$$Per(A) \leq \prod_{i=1}^{n} (r_i!)^{1/r_i}$$

- Corollary. If A is a row-stochastic matrix with maximal i-th row element $m_i \leq \frac{1}{r_i}$, where $r_i \in \mathbb{N}$, then

$$Per(A) \leq \left(\prod_{i=1}^{n} r_i\right)^{-1} \cdot \prod_{i=1}^{n} (r_i!)^{1/r_i}$$

- Since $r_i! \approx (2\pi r_i)^{1/r_i} \cdot \frac{r_i}{e}$, we see that for large r_i the upper bound we get is close to e^{-n}.

Strong concentration for balanced doubly stochastic matrices

- Let A be a doubly stochastic matrix with maximal entry t.

Strong concentration for balanced doubly stochastic matrices

- Let A be a doubly stochastic matrix with maximal entry t.

Then

$$e^{-n} \;<\; Per(A) \;\leq\; (2\pi/t)^{tn} \cdot e^{-n}$$

Strong concentration for balanced doubly stochastic matrices

- Let A be a doubly stochastic matrix with maximal entry t.

Then
$$e^{-n} \;<\; Per(A) \;\leq\; (2\pi/t)^{tn} \cdot e^{-n}$$

- If $t \leq O(\log(n)/n)$ then $e^{-n} < Per(A) \leq n^{O(\log(n))} \cdot e^{-n}$.

Strong concentration for balanced doubly stochastic matrices

- Let A be a doubly stochastic matrix with maximal entry t. Then

$$e^{-n} \;<\; Per(A) \;\leq\; (2\pi/t)^{tn} \cdot e^{-n}$$

- If $t \leq O(\log(n)/n)$ then $e^{-n} < Per(A) \leq n^{O(\log(n))} \cdot e^{-n}$.
- An application: A permanent of a random matrix is easy to approximate within a small factor.

Strong concentration for balanced doubly stochastic matrices

- Let A be a doubly stochastic matrix with maximal entry t. Then

$$e^{-n} \;<\; Per(A) \;\leq\; (2\pi/t)^{tn} \cdot e^{-n}$$

- If $t \leq O(\log(n)/n)$ then $e^{-n} < Per(A) \leq n^{O(\log(n))} \cdot e^{-n}$.
- An application: A permanent of a random matrix is easy to approximate within a small factor.
- I.e., if entries of A are independent standard exponential random variables

Strong concentration for balanced doubly stochastic matrices

- Let A be a doubly stochastic matrix with maximal entry t. Then

$$e^{-n} < Per(A) \leq (2\pi/t)^{tn} \cdot e^{-n}$$

- If $t \leq O(\log(n)/n)$ then $e^{-n} < Per(A) \leq n^{O(\log(n))} \cdot e^{-n}$.
- An application: A permanent of a random matrix is easy to approximate within a small factor.
- I.e., if entries of A are independent standard exponential random variables, then scaling approximates $Per(A)$ w.h.p. up to a factor of $n^{O(\log(n))}$.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = (\lambda_i a_{ij} \mu_j)$

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = (\lambda_i a_{ij} \mu_j)$ (this is the doubly stochastic part of A)

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = \left(\lambda_i a_{ij} \mu_j\right)$ (this is the doubly stochastic part of A) then $e^{-n} < Per(B) \leq n^{O(\log(n))} \cdot e^{-n}$.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = \left(\lambda_i a_{ij} \mu_j\right)$ (this is the doubly stochastic part of A) then $e^{-n} < Per(B) \leq n^{O(\log(n))} \cdot e^{-n}$.
- The entries of A have density e^{-t}, $t \geq 0$.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = (\lambda_i a_{ij} \mu_j)$ (this is the doubly stochastic part of A) then $e^{-n} < Per(B) \leq n^{O(\log(n))} \cdot e^{-n}$.
- The entries of A have density e^{-t}, $t \geq 0$. Their expectation is 1 and they are well concentrated.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = (\lambda_i a_{ij} \mu_j)$ (this is the doubly stochastic part of A) then $e^{-n} < Per(B) \leq n^{O(\log(n))} \cdot e^{-n}$.
- The entries of A have density e^{-t}, $t \geq 0$. Their expectation is 1 and they are well concentrated. Hence w.h.p. the row and the column sums are close to n.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = (\lambda_i a_{ij} \mu_j)$ (this is the doubly stochastic part of A) then $e^{-n} < Per(B) \leq n^{O(\log(n))} \cdot e^{-n}$.
- The entries of A have density e^{-t}, $t \geq 0$. Their expectation is 1 and they are well concentrated. Hence w.h.p. the row and the column sums are close to n. And, the maximal entry in each row is of order $\log(n)$.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = \left(\lambda_i a_{ij} \mu_j\right)$ (this is the doubly stochastic part of A) then $e^{-n} < Per(B) \leq n^{O(\log(n))} \cdot e^{-n}$.
- The entries of A have density e^{-t}, $t \geq 0$. Their expectation is 1 and they are well concentrated. Hence w.h.p. the row and the column sums are close to n. And, the maximal entry in each row is of order $\log(n)$.
- Therefore w.h.p. the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A are all about $1/\sqrt{n}$, and the maximal entry of B is at most $\log(n)/n$.

Proof

- Recall that we compute the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A and return $\left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1} \cdot e^{-n}$.
- It suffices to show that if $B = (\lambda_i a_{ij} \mu_j)$ (this is the doubly stochastic part of A) then $e^{-n} < Per(B) \leq n^{O(\log(n))} \cdot e^{-n}$.
- The entries of A have density e^{-t}, $t \geq 0$. Their expectation is 1 and they are well concentrated. Hence w.h.p. the row and the column sums are close to n. And, the maximal entry in each row is of order $\log(n)$.
- Therefore w.h.p. the scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ of A are all about $1/\sqrt{n}$, and the maximal entry of B is at most $\log(n)/n$.
- Done.

- Permanents.
- Contingency tables and integer flows.
- Mixed discriminants.

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums. (In this talk consider only square contingency tables.)

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.
- Counting the number $|\Sigma(R, C)|$ of contingency tables with row sums R and column sums C is interesting, because of applications in statistics, combinatorics, and other areas.

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.
- Counting the number $|\Sigma(R, C)|$ of contingency tables with row sums R and column sums C is interesting, because of applications in statistics, combinatorics, and other areas.
- Since exact counting is #-P hard DKM '94, the realistic goal is to approximate this number.

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.
- We want to approximate the number $|\Sigma(R, C)|$ of contingency tables with row sums R and column sums C.
- Connection to partition functions: Given a set of nonnegative weights $W = \{w_{ij}\}_{1 \le i,j \le n}$, let

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, \ldots, w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.
- We want to approximate the number $|\Sigma(R, C)|$ of contingency tables with row sums R and column sums C.
- Connection to partition functions: Given a set of nonnegative weights $W = \{w_{ij}\}_{1 \leq i,j \leq n}$, let

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, \ldots, w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.
- We want to approximate the number $|\Sigma(R, C)|$ of contingency tables with row sums R and column sums C.
- Connection to partition functions: Given a set of nonnegative weights $W = \{w_{ij}\}_{1 \leq i,j \leq n}$, let

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, ..., w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

- This is a multiset partition function B'16.

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.
- We want to approximate the number $|\Sigma(R,C)|$ of contingency tables with row sums R and column sums C.
- Connection to partition functions: Given a set of nonnegative weights $W = \{w_{ij}\}_{1 \leq i,j \leq n}$, let

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, \ldots, w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

- This is a multiset partition function B'16. Note that $|\Sigma(R,C)| = p_{\Sigma(R,C)}(1, \ldots, 1)$.

An application - counting contingency tables and integer flows

- A contingency table is an $n \times n$ integer matrix with prescribed row and column sums.
- Given a set of nonnegative weights $W = \{w_{ij}\}_{1 \le i,j \le n}$, let

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, \ldots, w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

- This is a multiset partition function B'16. Note that $|\Sigma(R,C)| = p_{\Sigma(R,C)}(1, \ldots, 1)$.
- For $0-1$ weights W, $p_{\Sigma(R,C)}(W)$ counts the number of integer flows in bipartite graphs.

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$.

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$. Let $N = \sum_i r_i = \sum_j c_j$.

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$. Let $N = \sum_i r_i = \sum_j c_j$.
- There are $\binom{\ell+s-1}{\ell-1}$ integer vectors of length ℓ and sum s.

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$. Let $N = \sum_i r_i = \sum_j c_j$.
- There are $\binom{\ell+s-1}{\ell-1}$ integer vectors of length ℓ and sum s.
- The probability that a random $n \times n$ integer matrix with entries summing to N has row sums $R = (r_1, ..., r_n)$ is

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$. Let $N = \sum_i r_i = \sum_j c_j$.
- There are $\binom{\ell+s-1}{\ell-1}$ integer vectors of length ℓ and sum s.
- The probability that a random $n \times n$ integer matrix with entries summing to N has row sums $R = (r_1, ..., r_n)$ is

$$\binom{N + n^2 - 1}{n^2 - 1}^{-1} \cdot \prod_{i=1}^{n} \binom{r_i + n - 1}{n - 1}$$

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$. Let $N = \sum_i r_i = \sum_j c_j$.
- The probability that a random $n \times n$ integer matrix with entries summing to N has row sums $R = (r_1, ..., r_n)$ is

$$\binom{N + n^2 - 1}{n^2 - 1}^{-1} \cdot \prod_{i=1}^{n} \binom{r_i + n - 1}{n - 1}$$

- Similarly for columns.

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$. Let $N = \sum_i r_i = \sum_j c_j$.
- The probability that a random $n \times n$ integer matrix with entries summing to N has row sums $R = (r_1, ..., r_n)$ is

$$\binom{N + n^2 - 1}{n^2 - 1}^{-1} \cdot \prod_{i=1}^{n} \binom{r_i + n - 1}{n - 1}$$

- Similarly for columns.
- Good's heuristic G'76: both events are essentially independent.

Good's heuristic for $|\Sigma(R, C)|$

- Let $R = (r_1, ..., r_n)$ and $C = (c_1, ..., c_n)$. Let $N = \sum_i r_i = \sum_j c_j$.
- The probability that a random $n \times n$ integer matrix with entries summing to N has row sums $R = (r_1, ..., r_n)$ is

$$\binom{N + n^2 - 1}{n^2 - 1}^{-1} \cdot \prod_{i=1}^{n} \binom{r_i + n - 1}{n - 1}$$

- Similarly for columns.
- Good's heuristic G'76: both events are essentially independent. Hence

$$|\Sigma(R, C)| \approx \binom{N + n^2 - 1}{n^2 - 1}^{-1} \cdot \prod_{i=1}^{n} \binom{r_i + n - 1}{n - 1} \cdot \prod_{j=1}^{n} \binom{c_j + n - 1}{n - 1}$$

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

$$\rho(R, C, W) \quad \leq \quad p_{\Sigma(R,C)}(W) \quad \leq \quad N^{\gamma n} \cdot \rho(R, C, W)$$

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

$$\rho(R, C, W) \quad \leq \quad p_{\Sigma(R,C)}(W) \quad \leq \quad N^{\gamma n} \cdot \rho(R, C, W),$$

where $N = \sum_i r_i = \sum_j c_j$.

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

$$\rho(R, C, W) \quad \leq \quad p_{\Sigma(R,C)}(W) \quad \leq \quad N^{\gamma n} \cdot \rho(R, C, W)$$

where $N = \sum_i r_i = \sum_j c_j$.

- This gives the best known rigorous approximation for $|\Sigma(R, C)|$ for many regimes of row and column sums.

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

$$\rho(R, C, W) \;\le\; p_{\Sigma(R,C)}(W) \;\le\; N^{\gamma n} \cdot \rho(R, C, W)$$

where $N = \sum_i r_i = \sum_j c_j$.

- This gives the best known rigorous approximation for $|\Sigma(R, C)|$ for many regimes of row and column sums.
- In particular, Good's heuristic is wrong.

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

$$\rho(R, C, W) \quad \leq \quad p_{\Sigma(R,C)}(W) \quad \leq \quad N^{\gamma n} \cdot \rho(R, C, W)$$

where $N = \sum_i r_i = \sum_j c_j$.

- This gives the best known rigorous approximation for $|\Sigma(R, C)|$ for many regimes of row and column sums.
- In particular, Good's heuristic is wrong. For typical margins R and C, the row and column events are positively correlated.

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

$$\rho(R,C,W) \;\leq\; p_{\Sigma(R,C)}(W) \;\leq\; N^{\gamma n} \cdot \rho(R,C,W)$$

where $N = \sum_i r_i = \sum_j c_j$.

- This gives the best known rigorous approximation for $|\Sigma(R,C)|$ for many regimes of row and column sums.
- In particular, Good's heuristic is wrong. For typical margins R and C, the row and column events are positively correlated.
- A key ingredient: strong concentration of the permanent for balanced doubly stochastic matrices.

Approximating the partition function

Barvinok'07-'16

- Theorem: There is an absolute constant $\gamma > 0$ and an efficiently computable quantity $\rho(R, C, W)$ such that

$$\rho(R, C, W) \;\; \leq \;\; p_{\Sigma(R,C)}(W) \;\; \leq \;\; N^{\gamma n} \cdot \rho(R, C, W)$$

where $N = \sum_i r_i = \sum_j c_j$.

- This gives the best known rigorous approximation for $|\Sigma(R, C)|$ for many regimes of row and column sums.
- In particular, Good's heuristic is wrong. For typical margins R and C, the row and column events are positively correlated.
- A key ingredient: strong concentration of the permanent for balanced doubly stochastic matrices. Will describe this for a very special case.

A very special case - magic squares

- A magic square is a contingency table in which all row and column sums equal a predetermined sum s.

A very special case - magic squares

- A magic square is a contingency table in which all row and column sums equal a predetermined sum s. The set of such magic squares is $\Sigma(s)$.

A very special case - magic squares

- A magic square is a contingency table in which all row and column sums equal a predetermined sum s. The set of such magic squares is $\Sigma(s)$.
- Note that $\Sigma(1) = S_n$.

A very special case - magic squares

- A magic square is a contingency table in which all row and column sums equal a predetermined sum s. The set of such magic squares is $\Sigma(s)$.
- Note that $\Sigma(1) = S_n$. Hence $p_{\Sigma(1)}(W) = Per(W)$.

A very special case - magic squares

- A magic square is a contingency table in which all row and column sums equal a predetermined sum s. The set of such magic squares is $\Sigma(s)$.
- Note that $\Sigma(1) = S_n$. Hence $p_{\Sigma(1)}(W) = Per(W)$.
- We describe an $n^{O(\log(n))}$-approximation algorithm for $|\Sigma(s)| = p_{\Sigma(s)}(1, ...1)$ from BLSY'07.

A very special case - magic squares

- A magic square is a contingency table in which all row and column sums equal a predetermined sum s. The set of such magic squares is $\Sigma(s)$.
- Note that $\Sigma(1) = S_n$. Hence $p_{\Sigma(1)}(W) = Per(W)$.
- We describe an $n^{O(\log(n))}$-approximation algorithm for $|\Sigma(s)| = p_{\Sigma(s)}(1, ...1)$ from BLSY'07.
- This algorithm, with minor changes, extends to balanced weights W.

A very special case - magic squares

- A magic square is a contingency table in which all row and column sums equal a predetermined sum s. The set of such magic squares is $\Sigma(s)$.
- Note that $\Sigma(1) = S_n$. Hence $p_{\Sigma(1)}(W) = Per(W)$.
- We describe an $n^{O(\log(n))}$-approximation algorithm for $|\Sigma(s)| = p_{\Sigma(s)}(1, ...1)$ from BLSY'07.
- This algorithm, with minor changes, extends to balanced weights W, when all the weights are within a constant factor from each other.

A very brief overview of the algorithm

- B'03: Represent $\Sigma(R, C)$ as an expectation of the permanent of a random matrix with exponentially distributed entries.

A very brief overview of the algorithm

- B'03: Represent $\Sigma(R, C)$ as an expectation of the permanent of a random matrix with exponentially distributed entries.
- Split corresponding integral into two parts - Good, corresponding to balanced matrices, and Bad.

A very brief overview of the algorithm

- B'03: Represent $\Sigma(R, C)$ as an expectation of the permanent of a random matrix with exponentially distributed entries.
- Split corresponding integral into two parts - Good, corresponding to balanced matrices, and Bad.
- Show that the bad part is negligible.

A very brief overview of the algorithm

- B'03: Represent $\Sigma(R, C)$ as an expectation of the permanent of a random matrix with exponentially distributed entries.
- Split corresponding integral into two parts - Good, corresponding to balanced matrices, and Bad.
- Show that the bad part is negligible.
- Compute the good part within a factor of $n^{O(\log(n))}$, using strong concentration of the permanent of the doubly stochastic part of the scaled matrix.

Representation $|\Sigma(n)|$ as an integral

- From now on restrict to a case of $n \times n$ magic squares with row and column sums n.

Representation $|\Sigma(n)|$ as an integral

- From now on restrict to a case of $n \times n$ magic squares with row and column sums n.
- Recall that for $n \times n$ matrices B and C, the tensor product $B \times C$ is the $n^2 \times n^2$ matrix

Representation $|\Sigma(n)|$ as an integral

- From now on restrict to a case of $n \times n$ magic squares with row and column sums n.
- Recall that for $n \times n$ matrices B and C, the tensor product $B \times C$ is the $n^2 \times n^2$ matrix

$$B \otimes C = \begin{pmatrix} b_{11}C & b_{12}C & \dots & b_{1n}C \\ b_{21}C & b_{22}C & \dots & b_{2n}C \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1}C & b_{n2}C & \dots & b_{nn}C \end{pmatrix}$$

Representation $|\Sigma(n)|$ as an integral

- From now on restrict to a case of $n \times n$ magic squares with row and column sums n.
- Recall that for $n \times n$ matrices B and C, the tensor product $B \times C$ is the $n^2 \times n^2$ matrix

$$B \otimes C \quad = \quad \begin{pmatrix} b_{11}C & b_{12}C & \dots & b_{1n}C \\ b_{21}C & b_{22}C & \dots & b_{2n}C \\ \vdots & \vdots & \vdots & \vdots \\ b_{n1}C & b_{n2}C & \dots & b_{nn}C \end{pmatrix}$$

- Bang's identity B'77, F'78:

$$Per(B \otimes J) = (n!)^{2n} \cdot \sum_{\alpha \in \Sigma(\bar{n})} \prod_{i,j=1}^{n} b_{i,j}^{\alpha_{ij}} / \alpha_{ij}!$$

Representation $|\Sigma(n)|$ as an integral

- From now on restrict to a case of $n \times n$ magic squares with row and column sums n.
- Bang's identity B'77, F'78:

$$Per(B \otimes J) = (n!)^{2n} \cdot \sum_{\alpha \in \Sigma(\bar{n})} \prod_{i,j=1}^{n} b_{i,j}^{\alpha_{ij}} / \alpha_{ij}!$$

- Let X be a standard exponential random variable (with density $f(t) = \exp\{-t\}$, $t \geq 0$).

Representation $|\Sigma(n)|$ as an integral

- From now on restrict to a case of $n \times n$ magic squares with row and column sums n.
- Bang's identity B'77, F'78:

$$Per(B \otimes J) = (n!)^{2n} \cdot \sum_{\alpha \in \Sigma(\bar{n})} \prod_{i,j=1}^{n} b_{i,j}^{\alpha_{ij}} / \alpha_{ij}!$$

- Let X be a standard exponential random variable (with density $f(t) = \exp\{-t\}$, $t \geq 0$). Then $\mathbb{E}X^k = k!$.

Representation $|\Sigma(n)|$ as an integral

- From now on restrict to a case of $n \times n$ magic squares with row and column sums n.
- Bang's identity B'77, F'78:

$$Per(B \otimes J) = (n!)^{2n} \cdot \sum_{\alpha \in \Sigma(\bar{n})} \prod_{i,j=1}^{n} b_{i,j}^{\alpha_{ij}} / \alpha_{ij}!$$

- Let X be a standard exponential random variable (with density $f(t) = \exp\{-t\}$, $t \geq 0$). Then $\mathbb{E}X^k = k!$.
- Hence B'03 (also in much higher generality)

$$|\Sigma(\bar{n})| \quad = \quad (n!)^{-2n} \cdot \int_{\mathbb{R}_+^{n^2}} Per(B \otimes J) d(B)$$

Representation $|\Sigma(n)|$ as an integral

- Bang's identity B'77, F'78:

$$Per(B \otimes J) = (n!)^{2n} \cdot \sum_{\alpha \in \Sigma(\bar{n})} \prod_{i,j=1}^{n} b_{i,j}^{\alpha_{ij}} / \alpha_{ij}!$$

- Let X be a standard exponential random variable (with density $f(t) = \exp\{-t\}$, $t \geq 0$). Then $\mathbb{E}X^k = k!$.
- Hence B'03

$$|\Sigma(\bar{n})| \;=\; (n!)^{-2n} \cdot \int_{\mathbb{R}_+^{n^2}} Per(B \otimes J) d(B) \;=$$

$$(n!)^{-2n} \cdot \int_{\mathbb{R}_+^{n^2}} Per(B \otimes J) \exp\{-\sum b_{i,j}\} db_{1,1} ... db_{n,n}$$

Estimating the integral

- Let $N = n^2$. For an $N \times N$ matrix A let $S(A)$ be the doubly stochastic part of A and $\sigma(A)$ be the inverse product of the scaling factors of A.

Estimating the integral

- Let $N = n^2$. For an $N \times N$ matrix A let $S(A)$ be the doubly stochastic part of A and $\sigma(A)$ be the inverse product of the scaling factors of A.
- Then $Per(A) = \sigma(A) \cdot Per(S(A))$.

Estimating the integral

- Let $N = n^2$. For an $N \times N$ matrix A let $S(A)$ be the doubly stochastic part of A and $\sigma(A)$ be the inverse product of the scaling factors of A.
- Then $Per(A) = \sigma(A) \cdot Per(S(A))$.
- Call A good if $e^{-N} \leq Per(S(A)) \leq N^{O(\log N)} \cdot e^{-N}$.

Estimating the integral

- Let $N = n^2$. For an $N \times N$ matrix A let $S(A)$ be the doubly stochastic part of A and $\sigma(A)$ be the inverse product of the scaling factors of A.
- Then $Per(A) = \sigma(A) \cdot Per(S(A))$.
- Call A good if $e^{-N} \leq Per(S(A)) \leq N^{O(\log N)} \cdot e^{-N}$. Otherwise A is bad.

Estimating the integral

- Let $N = n^2$. For an $N \times N$ matrix A let $S(A)$ be the doubly stochastic part of A and $\sigma(A)$ be the inverse product of the scaling factors of A.
- Then $Per(A) = \sigma(A) \cdot Per(S(A))$.
- Call A good if $e^{-N} \leq Per(S(A)) \leq N^{O(\log N)} \cdot e^{-N}$. Otherwise A is bad.
- Write

$$|\Sigma(n)| = \int_{\text{good } B} Per(B \otimes J) d(B) + \int_{\text{bad } B} Per(B \otimes J) d(B) =: I_g + I_b$$

Estimating the integral

- Call A good if $e^{-N} \leq Per(S(A)) \leq N^{O(\log N)} \cdot e^{-N}$. Otherwise A is bad.
- Write

$$|\Sigma(n)| = \int_{\text{good } B} Per(B \otimes J) d(B) + \int_{\text{bad } B} Per(B \otimes J) d(B) =: I_g + I_b$$

- We will argue that I_b is negligible and that I_g is easy to approximate.

The bad integral is small

- We know A is good if the maximal element of $S(A)$ is of order $O(\log(N)/N)$.

The bad integral is small

- We know A is good if the maximal element of $S(A)$ is of order $O(\log(N)/N)$.
- It is somewhat harder to guarantee properties of $S(B \otimes J)$ compared to $S(B)$ since tensor product amplifies bad events.

The bad integral is small

- We know A is good if the maximal element of $S(A)$ is of order $O(\log(N)/N)$.
- It is somewhat harder to guarantee properties of $S(B \otimes J)$ compared to $S(B)$ since tensor product amplifies bad events.
- Turns out to suffice to require that the sum of maximal row elements in $S(B \otimes J)$ is of order $O(\log N)$.

The bad integral is small

- We know A is good if the maximal element of $S(A)$ is of order $O\left(\log(N)/N\right)$.
- It is somewhat harder to guarantee properties of $S(B \otimes J)$ compared to $S(B)$ since tensor product amplifies bad events.
- Turns out to suffice to require that the sum of maximal row elements in $S(B \otimes J)$ is of order $O(\log N)$. And for this, it suffices w.h.p. that the sum of maximal row elements in $S(B)$ is of order $O(\log n)$.

The bad integral is small

- We know A is good if the maximal element of $S(A)$ is of order $O\left(\log(N)/N\right)$.
- It is somewhat harder to guarantee properties of $S(B \otimes J)$ compared to $S(B)$ since tensor product amplifies bad events.
- Turns out to suffice to require that the sum of maximal row elements in $S(B \otimes J)$ is of order $O(\log N)$. And for this, it suffices w.h.p. that the sum of maximal row elements in $S(B)$ is of order $O(\log n)$. Using large deviations.

The bad integral is small

- We know A is good if the maximal element of $S(A)$ is of order $O\left(\log(N)/N\right)$.
- It is somewhat harder to guarantee properties of $S(B \otimes J)$ compared to $S(B)$ since tensor product amplifies bad events.
- Turns out to suffice to require that the sum of maximal row elements in $S(B \otimes J)$ is of order $O(\log N)$. And for this, it suffices w.h.p. that the sum of maximal row elements in $S(B)$ is of order $O(\log n)$. Using large deviations.
- Bad integral

$$I_b = \int_{\text{bad } B} Per(B \otimes J)d(B)$$

is over bad matrices B - with large sum of maximal elements.

The bad integral is small

- Turns out to suffice to require that the sum of maximal row elements in $S(B \otimes J)$ is of order $O(\log N)$. And for this, it suffices w.h.p. that the sum of maximal row elements in $S(B)$ is of order $O(\log n)$. Using large deviations.
- Bad integral

$$I_b = \int_{\text{bad } B} Per(B \otimes J) d(B)$$

is over bad matrices B - with large sum of maximal elements.

- Turns out that I_b counts bad tables in $\Sigma(n)$ - these with specific structure: large sum of maximal elements.

The bad integral is small

- Turns out to suffice to require that the sum of maximal row elements in $S(B \otimes J)$ is of order $O(\log N)$. And for this, it suffices w.h.p. that the sum of maximal row elements in $S(B)$ is of order $O(\log n)$. Using large deviations.
- Bad integral

$$I_b = \int_{\text{bad } B} Per(B \otimes J) d(B)$$

is over bad matrices B - with large sum of maximal elements.

- Turns out that I_b counts bad tables in $\Sigma(n)$ - these with specific structure: large sum of maximal elements.
- The number of such tables is negligible via a combinatorial argument.

Approximating the good integral

- We have

$$|\Sigma(n)| \approx \int_{\text{good } B} Per(B \otimes J) d(B) =$$

$$\int_{\text{good } B} \sigma(B \otimes J) \cdot Per(S(B \otimes J)) d(B)$$

Approximating the good integral

- We have

$$|\Sigma(n)| \approx \int_{\text{good } B} Per(B \otimes J) d(B) =$$

$$\int_{\text{good } B} \sigma(B \otimes J) \cdot Per(S(B \otimes J)) d(B)$$

- For good B holds: $e^{-N} < Per(S(B \otimes J)) \leq N^{O(\log(N))} \cdot e^{-N}$.

Approximating the good integral

- We have

$$|\Sigma(n)| \approx \int_{\text{good } B} Per(B \otimes J) d(B) =$$

$$\int_{\text{good } B} \sigma(B \otimes J) \cdot Per(S(B \otimes J)) d(B)$$

- For good B holds: $e^{-N} < Per(S(B \otimes J)) \leq N^{O(\log(N))} \cdot e^{-N}$.
- Hence, up to a factor of $N^{O(\log(N))}$,

$$|\Sigma(n)| \approx e^{-N} \cdot \int_{\text{good } B} \sigma(B \otimes J) d(B)$$

Approximating the good integral

- We have

$$|\Sigma(n)| \approx \int_{\text{good } B} Per(B \otimes J) d(B) =$$

$$\int_{\text{good } B} \sigma(B \otimes J) \cdot Per(S(B \otimes J)) d(B)$$

- Hence, up to a factor of $N^{O(\log(N))}$,

$$|\Sigma(n)| \approx e^{-N} \cdot \int_{\text{good } B} \sigma(B \otimes J) d(B)$$

- Can we compute this?

Approximating the good integral

- We have

$$|\Sigma(n)| \approx \int_{\text{good } B} Per(B \otimes J) d(B) =$$

$$\int_{\text{good } B} \sigma(B \otimes J) \cdot Per(S(B \otimes J)) d(B)$$

- Hence, up to a factor of $N^{O(\log(N))}$,

$$|\Sigma(n)| \approx e^{-N} \cdot \int_{\text{good } B} \sigma(B \otimes J) d(B)$$

- Can we compute this? Yes, since $\sigma(B \otimes J)$ is log-concave in B B'05, G'06.

Log-concave functions

- A nonnegative function $f : \mathbb{R}^N \to \mathbb{R}$ is log-concave if

$$f(\lambda x + (1-\lambda)y) \quad \geq \quad f^{\lambda}(x) \cdot f^{1-\lambda}(y)$$

Log-concave functions

- A nonnegative function $f : \mathbb{R}^N \to \mathbb{R}$ is log-concave if

$$f(\lambda x + (1-\lambda) y) \quad \geq \quad f^{\lambda}(x) \cdot f^{1-\lambda}(y)$$

- E.g., the characteristic function of a convex body is log-concave.

Log-concave functions

- A nonnegative function $f : \mathbb{R}^N \to \mathbb{R}$ is log-concave if

$$f(\lambda x + (1-\lambda)y) \;\; \geq \;\; f^{\lambda}(x) \cdot f^{1-\lambda}(y)$$

- E.g., the characteristic function of a convex body is log-concave. And so is the exponential density $f(B) = \exp\{-\sum b_{i,j}\}$.

Log-concave functions

- A nonnegative function $f : \mathbb{R}^N \to \mathbb{R}$ is log-concave if

$$f(\lambda x + (1-\lambda)y) \;\; \geq \;\; f^{\lambda}(x) \cdot f^{1-\lambda}(y)$$

- E.g., the characteristic function of a convex body is log-concave. And so is the exponential density $f(B) = \exp\{-\sum b_{i,j}\}$.
- Log-concave functions are easy to integrate within an arbitrary error AK '91 (with application to computing volumes of convex bodies).

Log-concave functions

- A nonnegative function $f : \mathbb{R}^N \to \mathbb{R}$ is log-concave if

$$f(\lambda x + (1-\lambda)y) \geq f^{\lambda}(x) \cdot f^{1-\lambda}(y)$$

- E.g., the characteristic function of a convex body is log-concave. And so is the exponential density $f(B) = \exp\{-\sum b_{i,j}\}$.
- Log-concave functions are easy to integrate within an arbitrary error AK '91 (with application to computing volumes of convex bodies).
- Hence $\int_{\text{good } B} \sigma(B \otimes J) d(B)$ is computable.

Log-concave functions

- A nonnegative function $f : \mathbb{R}^N \to \mathbb{R}$ is log-concave if

$$f(\lambda x + (1-\lambda)y) \;\; \geq \;\; f^{\lambda}(x) \cdot f^{1-\lambda}(y)$$

- E.g., the characteristic function of a convex body is log-concave. And so is the exponential density $f(B) = \exp\{-\sum b_{i,j}\}$.
- Log-concave functions are easy to integrate within an arbitrary error AK '91 (with application to computing volumes of convex bodies).
- Hence $\int_{\text{good } B} \sigma(B \otimes J) d(B)$ is computable. Done.

The general contingency tables partition function

Barvinok '07-'16

- Use

The general contingency tables partition function

Barvinok ’07-’16

- Use
  - (Generalized) log-concavity of the scaling factor function.

The general contingency tables partition function

Barvinok '07-'16

- Use
  - (Generalized) log-concavity of the scaling factor function,
  - Strong concentration of the permanent for random 'exponential' matrices.

The general contingency tables partition function

Barvinok '07-'16

- Use
  - (Generalized) log-concavity of the scaling factor function,
  - Strong concentration of the permanent for random 'exponential' matrices,
  - The Prekopa-Leindler inequality.

The general contingency tables partition function

Barvinok '07-'16

- Use
  - (Generalized) log-concavity of the scaling factor function,
  - Strong concentration of the permanent for random 'exponential' matrices,
  - The Prekopa-Leindler inequality.
- For a structural result: approximate log-concavity of the partition function.

The general contingency tables partition function

Barvinok '07-'16

- A structural result: approximate log-concavity of the partition function.
- Recall the partition function with margins R and C and weights W:

The general contingency tables partition function

Barvinok '07-'16

- A structural result: approximate log-concavity of the partition function.
- Recall the partition function with margins R and C and weights W:

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, ..., w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

The general contingency tables partition function

Barvinok '07-'16

- A structural result: approximate log-concavity of the partition function.
- Recall the partition function with margins R and C and weights W:

$$p_{\Sigma(R,C)}(W) \;=\; p_{\Sigma(R,C)}(w_{11},\dots,w_{nn}) \;=\; \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

- If $R = \sum_i \lambda_i R_i$ and $C = \sum_i \lambda_i C_i$ (a convex combination) then

The general contingency tables partition function

Barvinok '07-'16

- A structural result: approximate log-concavity of the partition function.
- Recall the partition function with margins R and C and weights W:

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, \ldots, w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

- If $R = \sum_i \lambda_i R_i$ and $C = \sum_i \lambda_i C_i$ (a convex combination) then

$$p_{\Sigma(R,C)}(W) \geq \beta \cdot \prod_i p^{\lambda_i}_{\Sigma(R_i,C_i)}(W)$$

The general contingency tables partition function

Barvinok '07-'16

- A structural result: approximate log-concavity of the partition function.
- Recall the partition function with margins R and C and weights W:

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, \ldots, w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

- If $R = \sum_i \lambda_i R_i$ and $C = \sum_i \lambda_i C_i$ (a convex combination) then

$$p_{\Sigma(R,C)}(W) \geq \beta \cdot \prod_i p_{\Sigma(R_i,C_i)}^{\lambda_i}(W),$$

where β is not too small.

The general contingency tables partition function

Barvinok '07-'16

- Recall the partition function with margins R and C and weights W:

$$p_{\Sigma(R,C)}(W) = p_{\Sigma(R,C)}(w_{11}, \ldots, w_{nn}) = \sum_{D=(d_{ij})} \prod_{i,j=1}^{n} w_{ij}^{d_{ij}}$$

- If $R = \sum_i \lambda_i R_i$ and $C = \sum_i \lambda_i C_i$ (a convex combination) then

$$p_{\Sigma(R,C)}(W) \geq \beta \cdot \prod_i p_{\Sigma(R_i,C_i)}^{\lambda_i}(W),$$

where β is not too small.

- This and a subtle application of the capacity theory for polynomials (Gurvits) leads to approximation of general partition functions.

Convex optimization

Existence and properties of scaling factors

- Let $A - (a_{ij})$ be an $n \times n$ matrix with positive entries.

Convex optimization

Existence and properties of scaling factors

- Let $A - (a_{ij})$ be an $n \times n$ matrix with positive entries. We know that there exist $\{\lambda_i\}$ and $\{\mu_j\}$ such that $S(A) = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic.

Convex optimization

Existence and properties of scaling factors

- Let $A - (a_{ij})$ be an $n \times n$ matrix with positive entries. We know that there exist $\{\lambda_i\}$ and $\{\mu_j\}$ such that $S(A) = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
- We will give a new proof of this fact, with some nice side benefits, via convex optimization.

Convex optimization

Existence and properties of scaling factors

- Let $A - (a_{ij})$ be an $n \times n$ matrix with positive entries. We know that there exist $\{\lambda_i\}$ and $\{\mu_j\}$ such that $S(A) = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
- We will give a new proof of this fact, with some nice side benefits, via convex optimization.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.

Convex optimization

Existence and properties of scaling factors

- Let $A - (a_{ij})$ be an $n \times n$ matrix with positive entries. We know that there exist $\{\lambda_i\}$ and $\{\mu_j\}$ such that $S(A) = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
- We will give a new proof of this fact, with some nice side benefits, via convex optimization.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.
  - The function $\sigma(A) = \left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1}$ is log-concave.

Convex optimization

Existence and properties of scaling factors

- Let $A - (a_{ij})$ be an $n \times n$ matrix with positive entries. We know that there exist $\{\lambda_i\}$ and $\{\mu_j\}$ such that $S(A) = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
- We will give a new proof of this fact, with some nice side benefits, via convex optimization.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.
  - The function $\sigma(A) = \left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1}$ is log-concave.
  - This approach to scaling extends to other settings.

Convex optimization

Existence and properties of scaling factors

- Let $A - (a_{ij})$ be an $n \times n$ matrix with positive entries. We know that there exist $\{\lambda_i\}$ and $\{\mu_j\}$ such that $S(A) = (\lambda_i a_{ij} \mu_j)$ is doubly stochastic.
- We will give a new proof of this fact, with some nice side benefits, via convex optimization.
  - The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found efficiently.
  - The function $\sigma(A) = \left(\prod_i \lambda_i \cdot \prod_j \mu_j\right)^{-1}$ is log-concave.
  - This approach to scaling extends to other settings.
- The iterative approach of LSW'98 is strongly polynomial but seems to be limited to the matrix scaling case.

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A(x_1, \ldots, x_n) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$$

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$$

- f_A is strongly convex in $x = (x_1, ... x_n)$.

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A\left(x_1, ..., x_n\right) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$$

- f_A is strongly convex in $x = (x_1, ...x_n)$. That is, for $0 < \lambda < 1$ holds $f\left(\lambda x^{(1)} + (1-\lambda)x^{(2)}\right) < \lambda f\left(x^{(1)}\right) + (1-\lambda) f\left(x^{(2)}\right)$.

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$$

- f_A is strongly convex in $x = (x_1, ... x_n)$. That is, for $0 < \lambda < 1$ holds $f\left(\lambda x^{(1)} + (1-\lambda)x^{(2)}\right) < \lambda f\left(x^{(1)}\right) + (1-\lambda) f\left(x^{(2)}\right)$.
- In addition f_A tends to infinity if at least one of the x_i does.

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{n} a_{ij} e^{x_j} \right)$$

- f_A is strongly convex in $x = (x_1, ... x_n)$. That is, for $0 < \lambda < 1$ holds $f\left(\lambda x^{(1)} + (1-\lambda)x^{(2)}\right) < \lambda f\left(x^{(1)}\right) + (1-\lambda) f\left(x^{(2)}\right)$.
- In addition f_A tends to infinity if at least one of the x_i does.
- This means f_A has a unique minimum $x^* = \left(x_1^*, ... x_n^*\right)$ on $H = \{x_1 + ... + x_n = 0\}$.

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{n} a_{ij} e^{x_j} \right)$$

- f_A has a unique minimum $x^* = \left(x_1^*, ...x_n^*\right)$ on $H = \{x_1 + ... + x_n = 0\}$.
- The first order optimality conditions for x^* give

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{n} a_{ij} e^{x_j} \right)$$

- f_A has a unique minimum $x^* = (x_1^*, ...x_n^*)$ on $H = \{x_1 + ... + x_n = 0\}$.
- The first order optimality conditions for x^* give

$$\frac{\partial f_A}{\partial x_k} = \sum_{i=1}^{n} \frac{a_{ik} e^{x_k^*}}{\sum_{j=1}^{n} a_{ij} e^{x_j^*}} = \gamma$$

Scaling factors as a minimum of convex function

Following B'06

- Given A, let

$$f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{n} a_{ij} e^{x_j} \right)$$

- f_A has a unique minimum $x^* = (x_1^*, ...x_n^*)$ on $H = \{x_1 + ... + x_n = 0\}$.
- The first order optimality conditions for x^* give

$$\frac{\partial f_A}{\partial x_k} = \sum_{i=1}^{n} \frac{a_{ik} e^{x_k^*}}{\sum_{j=1}^{n} a_{ij} e^{x_j^*}} = \gamma,$$

for some constant γ and for all k.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, ..., x_n) = \sum_{i=1}^n \ln\left(\sum_{j=1}^n a_{ij} e^{x_j}\right)$.
- $x^* = \min_{x_1+...+x_n=0} f_A(x)$.
- The first order optimality conditions for x^* give

$$\frac{\partial f_A}{\partial x_k} = \sum_{i=1}^n \frac{a_{ik} e^{x_k^*}}{\sum_{j=1}^n a_{ij} e^{x_j^*}} = \gamma,$$

for some constant γ and for all k.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, ..., x_n) = \sum_{i=1}^n \ln\left(\sum_{j=1}^n a_{ij} e^{x_j}\right)$.
- $x^* = \min_{x_1+...+x_n=0} f_A(x)$.
- The first order optimality conditions for x^* give

$$\frac{\partial f_A}{\partial x_k} = \sum_{i=1}^n \frac{a_{ik} e^{x_k^*}}{\sum_{j=1}^n a_{ij} e^{x_j^*}} = \gamma,$$

for some constant γ and for all k.

- Let $\lambda_i = \left(\sum_{j=1}^n a_{ij} e^{x_j^*}\right)^{-1}$ and $\mu_j = e^{x_j^*}$.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, \ldots, x_n) = \sum_{i=1}^n \ln\left(\sum_{j=1}^n a_{ij} e^{x_j}\right)$.
- $x^* = \min_{x_1+\ldots+x_n=0} f_A(x)$.
- The first order optimality conditions for x^* give

$$\frac{\partial f_A}{\partial x_k} = \sum_{i=1}^{n} \frac{a_{ik} e^{x_k^*}}{\sum_{j=1}^n a_{ij} e^{x_j^*}} = \gamma,$$

for some constant γ and for all k.

- Let $\lambda_i = \left(\sum_{j=1}^n a_{ij} e^{x_j^*}\right)^{-1}$ and $\mu_j = e^{x_j^*}$. Let $B = (\lambda_i a_{ij} \mu_j)$.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, ..., x_n) = \sum_{i=1}^n \ln\left(\sum_{j=1}^n a_{ij} e^{x_j}\right)$.
- $x^* = \min_{x_1+...+x_n=0} f_A(x)$.
- The first order optimality conditions for x^* give

$$\frac{\partial f_A}{\partial x_k} = \sum_{i=1}^n \frac{a_{ik} e^{x_k^*}}{\sum_{j=1}^n a_{ij} e^{x_j^*}} = \gamma,$$

for some constant γ and for all k.

- Let $\lambda_i = \left(\sum_{j=1}^n a_{ij} e^{x_j^*}\right)^{-1}$ and $\mu_j = e^{x_j^*}$. Let $B = (\lambda_i a_{ij} \mu_j)$.
- Each column sum in B is 1 and each row sum is γ.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$.
- $x^* = \min_{x_1+...+x_n=0} f_A(x)$.
- The first order optimality conditions for x^* give

$$\frac{\partial f_A}{\partial x_k} = \sum_{i=1}^{n} \frac{a_{ik} e^{x_k^*}}{\sum_{j=1}^{n} a_{ij} e^{x_j^*}} = \gamma,$$

for some constant γ and for all k.

- Let $\lambda_i = \left(\sum_{j=1}^{n} a_{ij} e^{x_j^*}\right)^{-1}$ and $\mu_j = e^{x_j^*}$. Let $B = (\lambda_i a_{ij} \mu_j)$.
- Each column sum in B is 1 and each row sum is γ. Hence $\gamma = 1$ and B is doubly stochastic.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$.
- $x^* = \min_{x_1+...+x_n=0} f_A(x)$.
- Let $\lambda_i = \left(\sum_{j=1}^{n} a_{ij} e^{x_j^*}\right)^{-1}$ and $\mu_j = e^{x_j^*}$. Let $B = (\lambda_i a_{ij} \mu_j)$.
- B is doubly stochastic.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$.
- $x^* = \min_{x_1+...+x_n=0} f_A(x)$.
- Let $\lambda_i = \left(\sum_{j=1}^{n} a_{ij} e^{x_j^*}\right)^{-1}$ and $\mu_j = e^{x_j^*}$. Let $B = (\lambda_i a_{ij} \mu_j)$.
- B is doubly stochastic.
- The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found using the ellipsoid algorithm.

Scaling factors as a minimum of convex function

Following B'06

- $f_A(x_1, ..., x_n) = \sum_{i=1}^{n} \ln \left(\sum_{j=1}^{n} a_{ij} e^{x_j} \right)$.
- $x^* = \min_{x_1+...+x_n=0} f_A(x)$.
- Let $\lambda_i = \left(\sum_{j=1}^{n} a_{ij} e^{x_j^*} \right)^{-1}$ and $\mu_j = e^{x_j^*}$. Let $B = (\lambda_i a_{ij} \mu_j)$.
- B is doubly stochastic.
- The scaling factors $\{\lambda_i\}$ and $\{\mu_j\}$ can be found using the ellipsoid algorithm, attaining (arbitrary) precision ϵ in time *Poly* $(n, \log(1/\epsilon), \nu)$, where $\nu = \log\left(\max\{a_{ij}\} / \min\{a_{ij}\}\right)$.

Log-concavity of $\sigma(A)$

Following B'06

- We have

$$\sigma(A) = \left(\prod_i \lambda_i\right)^{-1} \cdot \left(\prod_j \mu_j\right)^{-1}$$

Log-concavity of $\sigma(A)$

Following B'06

- We have

$$\sigma(A) = \left(\prod_i \lambda_i\right)^{-1} \cdot \left(\prod_j \mu_j\right)^{-1} = \left(\prod_{i=1}^{n}\sum_{j=1}^{n} a_{ij} e^{x_j^*}\right) \cdot e^{-\sum_j x_j^*}$$

Log-concavity of $\sigma(A)$

Following B'06

- We have

$$\sigma(A) = \left(\prod_i \lambda_i\right)^{-1} \cdot \left(\prod_j \mu_j\right)^{-1} = \left(\prod_{i=1}^{n} \sum_{j=1}^{n} a_{ij} e^{x_j^*}\right) \cdot e^{-\sum_j x_j^*} =$$

$$\prod_{i=1}^{n} \sum_{j=1}^{n} a_{ij} e^{x_j^*}$$

Log-concavity of $\sigma(A)$

Following B'06

- We have

$$\sigma(A) = \left(\prod_i \lambda_i\right)^{-1} \cdot \left(\prod_j \mu_j\right)^{-1} = \left(\prod_{i=1}^n \sum_{j=1}^n a_{ij} e^{x_j^*}\right) \cdot e^{-\sum_j x_j^*} =$$

$$\prod_{i=1}^n \sum_{j=1}^n a_{ij} e^{x_j^*} = e^{f_A(x^*)}$$

- Hence $\ln \sigma(A) = f_A(x^*) = \min_{x_1+\ldots+x_n=0} f_A(x)$.

Log-concavity of $\sigma(A)$

Following B'06

- $\ln \sigma(A) = f_A(x^*) = \min_{x_1+\ldots+x_n=0} f_A(x)$.
- For a fixed x, $f_A(x) = \sum_{i=1}^n \ln\left(\sum_{j=1}^n a_{ij} e^{x_j}\right)$ is concave in A.

Log-concavity of $\sigma(A)$

Following B'06

- $\ln \sigma(A) = f_A(x^*) = \min_{x_1+\ldots+x_n=0} f_A(x)$.
- For a fixed x, $f_A(x) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$ is concave in A.
- Hence, for x such that $x_1 + \ldots + x_n = 0$,

$$f_{\lambda A_1 + (1-\lambda) A_2}(x) \;\geq\; \lambda f_{A_1}(x) + (1-\lambda) f_{A_2}(x)$$

Log-concavity of $\sigma(A)$

Following B'06

- $\ln \sigma(A) = f_A(x^*) = \min_{x_1+\ldots+x_n=0} f_A(x)$.
- For a fixed x, $f_A(x) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$ is concave in A.
- Hence, for x such that $x_1 + \ldots + x_n = 0$,

$$f_{\lambda A_1 + (1-\lambda) A_2}(x) \;\geq\; \lambda f_{A_1}(x) + (1-\lambda) f_{A_2}(x) \;\geq$$

$$\lambda \ln \sigma(A_1) + (1-\lambda) \ln \sigma(A_2)$$

Log-concavity of $\sigma(A)$

Following B'06

- $\ln \sigma(A) = f_A(x^*) = \min_{x_1+\ldots+x_n=0} f_A(x)$.
- For a fixed x, $f_A(x) = \sum_{i=1}^n \ln\left(\sum_{j=1}^n a_{ij} e^{x_j}\right)$ is concave in A.
- Hence, for x such that $x_1 + \ldots + x_n = 0$,

$$f_{\lambda A_1+(1-\lambda)A_2}(x) \;\geq\; \lambda f_{A_1}(x) + (1-\lambda) f_{A_2}(x) \;\geq$$

$$\lambda \ln \sigma(A_1) + (1-\lambda) \ln \sigma(A_2)$$

- Minimizing for x:

$\ln \sigma(\lambda A_1 + (1-\lambda) A_2) \geq \lambda \ln \sigma(A_1) + (1-\lambda) \ln \sigma(A_2)$.

Log-concavity of $\sigma(A)$

Following B'06

- $\ln \sigma(A) = f_A(x^*) = \min_{x_1+\ldots+x_n=0} f_A(x)$.
- For a fixed x, $f_A(x) = \sum_{i=1}^{n} \ln\left(\sum_{j=1}^{n} a_{ij} e^{x_j}\right)$ is concave in A.
- Hence, for x such that $x_1 + \ldots + x_n = 0$,

$$f_{\lambda A_1+(1-\lambda)A_2}(x) \; \geq \; \lambda f_{A_1}(x) + (1-\lambda) f_{A_2}(x) \; \geq$$

$$\lambda \ln \sigma(A_1) + (1-\lambda) \ln \sigma(A_2)$$

- Minimizing for x:
$\ln \sigma(\lambda A_1 + (1-\lambda)A_2) \geq \lambda \ln \sigma(A_1) + (1-\lambda) \ln \sigma(A_2)$.
- Done.

- Permanents.
- Contingency tables and integer flows.
- Mixed discriminants.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, \ldots A_n$ be symmetric $n \times n$ matrices.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
- $$D(A_1, ..., A_n) = \text{coef}_{x_1,...,x_n} p(x_1, ..., x_n)$$

  is the mixed discriminant of $A_1, ..., A_n$.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
-
$$D(A_1, ..., A_n) = \text{coef}_{x_1,...,x_n} p(x_1, ..., x_n)$$
is the mixed discriminant of $A_1, ..., A_n$.
- Examples.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
- $$D(A_1, ..., A_n) \quad = \quad \text{coef}_{x_1, ..., x_n} p(x_1, ..., x_n)$$

  is the mixed discriminant of $A_1, ..., A_n$.
- Examples.
  - $D(A, ..., A) = n! \cdot Det(A)$.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
- $$D(A_1, ..., A_n) = \text{coef}_{x_1, ..., x_n} p(x_1, ..., x_n)$$

  is the mixed discriminant of $A_1, ..., A_n$.
- Examples.
  - $D(A, ..., A) = n! \cdot Det(A)$. Easily computable.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
-
$$D(A_1, ..., A_n) = \text{coef}_{x_1,...,x_n} p(x_1, ..., x_n)$$
is the mixed discriminant of $A_1, ..., A_n$.
- Examples.
  - $D(A, ..., A) = n! \cdot Det(A)$. Easily computable.
  - If $A_1, ..., A_n$ are diagonal matrices with diagonals $a_1, ...a_n$, then $D(A_1, ..., A_n) = Per(A)$, where the columns of A are $a_1, ...a_n$.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
- $$D(A_1, ..., A_n) = \text{coef}_{x_1,...,x_n} p(x_1, ..., x_n)$$

  is the mixed discriminant of $A_1, ..., A_n$.
- Examples.
  - $D(A, ..., A) = n! \cdot Det(A)$. Easily computable.
  - If $A_1, ..., A_n$ are diagonal matrices with diagonals $a_1, ...a_n$, then $D(A_1, ..., A_n) = Per(A)$, where the columns of A are $a_1, ...a_n$. Hard to compute.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
- $$D(A_1, ..., A_n) = \text{coef}_{x_1,...,x_n} p(x_1, ..., x_n)$$

  is the mixed discriminant of $A_1, ..., A_n$.
- Examples.
  - $D(A, ..., A) = n! \cdot Det(A)$. Easily computable.
  - If $A_1, ..., A_n$ are diagonal matrices with diagonals $a_1, ...a_n$, then $D(A_1, ..., A_n) = Per(A)$, where the columns of A are $a_1, ...a_n$. Hard to compute. Easy to approximate.

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
- $$D(A_1, ..., A_n) = \text{coef}_{x_1,...,x_n} p(x_1, ..., x_n)$$
  is the mixed discriminant of $A_1, ..., A_n$.
- Explicitly (as a partition function): Let $A_k = \left(a_{ij}^{(k)}\right)$. Then

Higher-dimensional partition functions - mixed discriminants

- Let $A_1, ...A_n$ be symmetric $n \times n$ matrices. Then $p(x_1, ..., x_n) = Det(x_1 A_1 + ... + x_n A_n)$ is a homogeneous degree-n polynomial in $\{x_i\}$.
- $$D(A_1, ..., A_n) = \text{coef}_{x_1,...,x_n} p(x_1, ..., x_n)$$

  is the mixed discriminant of $A_1, ..., A_n$.
- Explicitly (as a partition function): Let $A_k = \left(a_{ij}^{(k)}\right)$. Then

$$D(A_1, ..., A_n) = \sum_{\sigma,\tau \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}^{(\tau(i))}$$

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) \quad = \quad \mathrm{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

$(x_1, ..., x_n \geq 0)$ is the mixed volume of $K_1, ..., K_n$.

- Mixed volumes are important.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, \dots, K_n$ in $\mathbb{R}^n$

$$V(K_1, \dots, K_n) \quad = \quad \mathrm{coef}_{x_1,\dots,x_n} V(x_1 K_1 + \dots + x_n K_n)$$

$(x_1, \dots, x_n \geq 0)$ is the mixed volume of $K_1, \dots, K_n$.

- Mixed volumes are important.
  - $V(K, \dots, K) = n! \cdot V(K)$.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) \quad = \quad \mathrm{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

$(x_1, ..., x_n \geq 0)$ is the mixed volume of $K_1, ..., K_n$.

- Mixed volumes are important.
  - $V(K, ..., K) = n! \cdot V(K)$. Easy to approximate.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) \quad = \quad \mathrm{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

$(x_1, ..., x_n \geq 0)$ is the mixed volume of $K_1, ..., K_n$.

- Mixed volumes are important.
  - $V(K, ..., K) = n! \cdot V(K)$. Easy to approximate.
  - Mixed volume of axis-parallel boxes is the permanent of the appropriate matrix.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) \;=\; \text{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

$(x_1, ..., x_n \geq 0)$ is the mixed volume of $K_1, ..., K_n$.

- Mixed volumes are important.
  - $V(K, ..., K) = n! \cdot V(K)$. Easy to approximate.
  - Mixed volume of axis-parallel boxes is the permanent of the appropriate matrix. Hard to compute, easy to approximate.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) = \text{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

$(x_1, ..., x_n \geq 0)$ is the mixed volume of $K_1, ..., K_n$.

- Mixed volumes are important.
  - $V(K, ..., K) = n! \cdot V(K)$. Easy to approximate.
  - Mixed volume of axis-parallel boxes is the permanent of the appropriate matrix. Hard to compute, easy to approximate.
  - There is more to it:

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) \quad = \quad \text{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

$(x_1, ..., x_n \geq 0)$ is the mixed volume of $K_1, ..., K_n$.

- Mixed volumes are important.
  - $V(K, ..., K) = n! \cdot V(K)$. Easy to approximate.
  - Mixed volume of axis-parallel boxes is the permanent of the appropriate matrix. Hard to compute, easy to approximate.
  - There is more to it: Let B be the unit ball. Then $V(K, ..., K, B)$ is proportional to the surface area of K.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) = \text{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

($x_1, ..., x_n \geq 0$) is the mixed volume of $K_1, ..., K_n$.

- Mixed volumes are important.
  - $V(K, ..., K) = n! \cdot V(K)$. Easy to approximate.
  - Mixed volume of axis-parallel boxes is the permanent of the appropriate matrix. Hard to compute, easy to approximate.
  - There is more to it: Let B be the unit ball. Then $V(K, ..., K, B)$ is proportional to the surface area of K. Etc.

Mixed discriminants and mixed volumes

- Mixed discriminants were introduced by A'38 in work on mixed volumes. For convex bodies $K_1, ..., K_n$ in $\mathbb{R}^n$

$$V(K_1, ..., K_n) \quad = \quad \text{coef}_{x_1,...,x_n} V(x_1 K_1 + ... + x_n K_n)$$

$(x_1, ..., x_n \geq 0)$ is the mixed volume of $K_1, ..., K_n$.

  - $V(K, ..., K) = n! \cdot V(K)$. Easy to approximate.
  - Mixed volume of axis-parallel boxes is the permanent of the appropriate matrix. Hard to compute, easy to approximate.
  - There is more to it: Let B be the unit ball. Then $V(K, ..., K, B)$ is proportional to the surface area of K. Etc.
  - The best known approximation factor is $n^{O(n)}$ B'97.

Deterministic approximation of mixed discriminant and mixed volume

- DGH'98: Is there a deterministic algorithm to approximate mixed volume?

Deterministic approximation of mixed discriminant and mixed volume

- DGH'98: Is there a deterministic algorithm to approximate mixed volume? A lower bound of $(n/\log(n))^{O(n)}$ for deterministic volume approximation BF'87.

Deterministic approximation of mixed discriminant and mixed volume

- DGH'98: Is there a deterministic algorithm to approximate mixed volume? A lower bound of $(n/\log(n))^{O(n)}$ for deterministic volume approximation BF'87.
- B'97: Reduction to approximating mixed discriminants of positive semidefinite matrices, within loss of $n^{O(n)}$.

Deterministic approximation of mixed discriminant and mixed volume

- DGH'98: Is there a deterministic algorithm to approximate mixed volume? A lower bound of $(n/\log(n))^{O(n)}$ for deterministic volume approximation BF'87.
- B'97: Reduction to approximating mixed discriminants of positive semidefinite matrices, within loss of $n^{O(n)}$. A probabilistic algorithm to approximate mixed discriminants and hence mixed volumes.

Deterministic approximation of mixed discriminant and mixed volume

- DGH'98: Is there a deterministic algorithm to approximate mixed volume? A lower bound of $(n/\log(n))^{O(n)}$ for deterministic volume approximation BF'87.
- B'97: Reduction to approximating mixed discriminants of positive semidefinite matrices, within loss of $n^{O(n)}$. A probabilistic algorithm to approximate mixed discriminants and hence mixed volumes.
- GS'01: A deterministic algorithm to approximate mixed discriminants (of psd matrices) up to a factor of e^n using scaling.

Scaling n-tuples of matrices

- Scaling how?

Scaling n-tuples of matrices

- Scaling how? Need to keep track of mixed discriminant.

Scaling n-tuples of matrices

- Scaling how? Need to keep track of mixed discriminant.
- Scaling to what?

Scaling n-tuples of matrices

- Scaling how? Need to keep track of mixed discriminant.
- Scaling to what? Need to define a goal object, that will be sufficiently balanced so that mixed discriminant is concentrated.

Scaling n-tuples of matrices

- Scaling how? Need to keep track of mixed discriminant.
- Scaling to what? Need to define a goal object, that will be sufficiently balanced so that mixed discriminant is concentrated.
- The second step will require proving upper and lower bounds on the mixed discriminant.

Allowed scaling operations

- $D(A_1, ..., A_n) = \sum_{\sigma,\tau \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}^{(\tau(i))}$ is multi-linear, so "row" and "column" scaling should work.

Allowed scaling operations

- $D(A_1, ..., A_n) = \sum_{\sigma,\tau \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}^{(\tau(i))}$ is multi-linear, so "row" and "column" scaling should work.
- Scaling factors: Let $A_1, ..., A_n$ be psd matrices. Let Λ be a psd matrix and let $\{\mu_j\}$ be nonnegative numbers.

Allowed scaling operations

- $D(A_1, ..., A_n) = \sum_{\sigma,\tau \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}^{(\tau(i))}$ is multi-linear, so "row" and "column" scaling should work.
- Scaling factors: Let $A_1, ..., A_n$ be psd matrices. Let Λ be a psd matrix and let $\{\mu_j\}$ be nonnegative numbers. The matrices $B_i = \mu_i \Lambda^{1/2} A_i \Lambda^{1/2}$ are psd, and

Allowed scaling operations

- $D(A_1, ..., A_n) = \sum_{\sigma,\tau \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}^{(\tau(i))}$ is multi-linear, so "row" and "column" scaling should work.
- Scaling factors: Let $A_1, ..., A_n$ be psd matrices. Let Λ be a psd matrix and let $\{\mu_j\}$ be nonnegative numbers. The matrices $B_i = \mu_i \Lambda^{1/2} A_i \Lambda^{1/2}$ are psd, and

$$D(B_1, ..., B_n) = Det(\Lambda) \cdot \prod_j \mu_j \cdot D(A_1, ..., A_n)$$

Allowed scaling operations

- $D(A_1, ..., A_n) = \sum_{\sigma,\tau \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}^{(\tau(i))}$ is multi-linear, so "row" and "column" scaling should work.
- Scaling factors: Let $A_1, ..., A_n$ be psd matrices. Let Λ be a psd matrix and let $\{\mu_j\}$ be nonnegative numbers. The matrices $B_i = \mu_i \Lambda^{1/2} A_i \Lambda^{1/2}$ are psd, and

$$D(B_1, ..., B_n) = Det(\Lambda) \cdot \prod_j \mu_j \cdot D(A_1, ..., A_n)$$

- In fact, even a more general scaling operation - multiplying on the right and on the left by different matrices - is allowed.

Allowed scaling operations

- $D(A_1, ..., A_n) = \sum_{\sigma,\tau \in S_n} sign(\sigma) \prod_{i=1}^{n} a_{i\sigma(i)}^{(\tau(i))}$ is multi-linear, so "row" and "column" scaling should work.
- Scaling factors: Let $A_1, ..., A_n$ be psd matrices. Let Λ be a psd matrix and let $\{\mu_j\}$ be nonnegative numbers. The matrices $B_i = \mu_i \Lambda^{1/2} A_i \Lambda^{1/2}$ are psd, and

$$D(B_1, ..., B_n) = Det(\Lambda) \cdot \prod_j \mu_j \cdot D(A_1, ..., A_n)$$

- In fact, even a more general scaling operation - multiplying on the right and on the left by different matrices - is allowed. E.g., this allows to transform any commuting n-tuple of matrices to diagonal matrices, reducing the computation to computing a permanent.

Allowed scaling operations

- Scaling factors: Let $A_1, ..., A_n$ be psd matrices. Let Λ be a psd matrix and let $\{\mu_j\}$ be nonnegative numbers. The matrices $B_i = \mu_i \Lambda^{1/2} A_i \Lambda^{1/2}$ are psd, and

$$D(B_1, ..., B_n) = Det(\Lambda) \cdot \prod_j \mu_j \cdot D(A_1, ..., A_n)$$

- In fact, even a more general scaling operation - multiplying on the right and on the left by different matrices - is allowed. E.g., this allows to transform any commuting n-tuple of matrices to diagonal matrices, reducing the computation to computing a permanent.
- In this way mixed discriminant is a non-commutative generalization of permanent.

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, ..., B_n$ of psd matrices is doubly stochastic if

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, ..., B_n$ of psd matrices is doubly stochastic if
  - The trace of each matrix is 1.

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, ..., B_n$ of psd matrices is doubly stochastic if
  - The trace of each matrix is 1.
  - The sum of the matrices is the identity matrix.

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, ..., B_n$ of psd matrices is doubly stochastic if
  - The trace of each matrix is 1.
  - The sum of the matrices is the identity matrix.
- The commutative case corresponds to a usual doubly stochastic matrix.

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, \ldots, B_n$ of psd matrices is doubly stochastic if
  - The trace of each matrix is 1.
  - The sum of the matrices is the identity matrix.
- An n-tuple $A_1, \ldots, A_n$ is positive if for any $S \subset [n]$, $|S| < n$, holds $\text{rank}\left(\sum_{i \in S} A_i\right) > |S|$.

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, ..., B_n$ of psd matrices is doubly stochastic if
  - The trace of each matrix is 1.
  - The sum of the matrices is the identity matrix.
- An n-tuple $A_1, ..., A_n$ is positive if for any $S \subset [n]$, $|S| < n$, holds $\text{rank}\left(\sum_{i \in S} A_i\right) > |S|$.
- GS'01: Let $\{A_i\}$ be a positive n-tuple of psd matrices. Then

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, ..., B_n$ of psd matrices is doubly stochastic if
  - The trace of each matrix is 1.
  - The sum of the matrices is the identity matrix.

- An n-tuple $A_1, ..., A_n$ is positive if for any $S \subset [n]$, $|S| < n$, holds $\text{rank}\left(\sum_{i \in S} A_i\right) > |S|$.
- GS'01: Let $\{A_i\}$ be a positive n-tuple of psd matrices. Then
  - There exist scaling factors Λ and $\{\mu_j\}$ such that $\{B_i = \mu_i \Lambda^{1/2} A_i \Lambda^{1/2}\}$ is doubly stochastic.

Scaling to a doubly stochastic tuple

- An n-tuple $B_1, ..., B_n$ of psd matrices is doubly stochastic if
  - The trace of each matrix is 1.
  - The sum of the matrices is the identity matrix.

- An n-tuple $A_1, ..., A_n$ is positive if for any $S \subset [n]$, $|S| < n$, holds $\text{rank}\left(\sum_{i \in S} A_i\right) > |S|$.
- GS'01: Let $\{A_i\}$ be a positive n-tuple of psd matrices. Then
  - There exist scaling factors Λ and $\{\mu_j\}$ such that $\{B_i = \mu_i \Lambda^{1/2} A_i \Lambda^{1/2}\}$ is doubly stochastic.
  - The scaling factors can be found efficiently.

Proof - via convex optimization

- Let $A = \{A_i\}$ be a positive n-tuple of psd matrices.

Proof - via convex optimization

- Let $A = \{A_i\}$ be a positive n-tuple of psd matrices. Let

$$f_A(x_1, ..., x_n) \quad = \quad \ln Det\left(e^{x_1} A_1 + ... + e^{x_n} A_n\right)$$

Proof - via convex optimization

- Let $A = \{A_i\}$ be a positive n-tuple of psd matrices. Let

$$f_A(x_1, ..., x_n) \quad = \quad \ln Det(e^{x_1} A_1 + ... + e^{x_n} A_n)$$

- f_A is strongly convex in $x = (x_1, ...x_n)$.

Proof - via convex optimization

- Let $A = \{A_i\}$ be a positive n-tuple of psd matrices. Let

$$f_A(x_1, ..., x_n) \;=\; \ln Det\,(e^{x_1}A_1 + ... + e^{x_n}A_n)$$

- f_A is strongly convex in $x = (x_1, ...x_n)$.
- In addition f_A tends to infinity if at least one of the x_i does.

Proof - via convex optimization

- Let $A = \{A_i\}$ be a positive n-tuple of psd matrices. Let

$$f_A(x_1, ..., x_n) \;=\; \ln Det\left(e^{x_1}A_1 + ... + e^{x_n}A_n\right)$$

- f_A is strongly convex in $x = (x_1, ...x_n)$.
- In addition f_A tends to infinity if at least one of the x_i does.

This means that f_A has a unique minimum $x^* = \left(x_1^*, ...x_n^*\right)$ on $H = \{x_1 + ... + x_n = 0\}$.

Proof - via convex optimization

- Let $A = \{A_i\}$ be a positive n-tuple of psd matrices. Let

$$f_A(x_1, ..., x_n) \quad = \quad \ln Det\,(e^{x_1} A_1 + ... + e^{x_n} A_n)$$

- f_A is strongly convex in $x = (x_1, ...x_n)$.
- In addition f_A tends to infinity if at least one of the x_i does.

This means that f_A has a unique minimum $x^* = \left(x_1^*, ...x_n^*\right)$ on $H = \{x_1 + ... + x_n = 0\}$.

- The first order optimality conditions on x^* imply the existence of the scaling factors for A.

Proof - via convex optimization

- Let $A = \{A_i\}$ be a positive n-tuple of psd matrices. Let

$$f_A(x_1, ..., x_n) \quad = \quad \ln Det\,(e^{x_1}A_1 + ... + e^{x_n}A_n)$$

- f_A is strongly convex in $x = (x_1, ...x_n)$.
- In addition f_A tends to infinity if at least one of the x_i does. This means that f_A has a unique minimum $x^* = \left(x_1^*, ...x_n^*\right)$ on $H = \{x_1 + ... + x_n = 0\}$.
- The first order optimality conditions on x^* imply the existence of the scaling factors for A.
- They can be found efficiently (but not strongly polynomially) using the ellipsoid method.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.
- For a doubly stochastic n-tuple $A = A_1, ..., A_n$ of psd matrices

$$D(A) \geq \frac{n!}{n^n} > e^{-n}$$

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.
- For a doubly stochastic n-tuple $A = A_1, ..., A_n$ of psd matrices

$$D(A) \geq \frac{n!}{n^n} > e^{-n}$$

- Difficult.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.
- For a doubly stochastic n-tuple $A = A_1, ..., A_n$ of psd matrices

$$D(A) \geq \frac{n!}{n^n} > e^{-n}$$

- Difficult. Has been stated as a conjecture by B'89.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.
- For a doubly stochastic n-tuple $A = A_1, ..., A_n$ of psd matrices

$$D(A) \geq \frac{n!}{n^n} > e^{-n}$$

- Difficult. Has been stated as a conjecture by B'89. Proved in G'01 emulating the lower bound proof for permanents.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.
- For a doubly stochastic n-tuple $A = A_1, ..., A_n$ of psd matrices

$$D(A) \geq \frac{n!}{n^n} > e^{-n}$$

- Difficult. Has been stated as a conjecture by B'89. Proved in G'01 emulating the lower bound proof for permanents. Done.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.
- For a doubly stochastic n-tuple $A = A_1, ..., A_n$ of psd matrices

$$D(A) \geq \frac{n!}{n^n} > e^{-n}$$

- Difficult. Has been stated as a conjecture by B'89. Proved in G'01 emulating the lower bound proof for permanents. Done.
- Following this, Leonid Gurvits developed a new general approach to lower bound coefficients of H-stable polynomials via their capacity.

Concentration of mixed discriminant for doubly stochastic tuples

- For any n-tuple $A = A_1, ..., A_n$ of psd matrices $D(A) = D(A_1, ..., A_n) \leq Det(A_1 + ...A_n)$. Hence if A is doubly stochastic, $D(A) \leq 1$. Easy.
- For a doubly stochastic n-tuple $A = A_1, ..., A_n$ of psd matrices

$$D(A) \geq \frac{n!}{n^n} > e^{-n}$$

- Difficult. Has been stated as a conjecture by B'89. Proved in G'01 emulating the lower bound proof for permanents. Done.
- Following this, Leonid Gurvits developed a new general approach to lower bound coefficients of H-stable polynomials via their capacity. This led to easy proofs for the permanent and mixed discriminant lower bounds and other results.

Rado's theorem with volume

A geometric corollary

- $V_1, ..., V_n$ are families of vectors in $\mathbb{R}^n$.

Rado's theorem with volume

A geometric corollary

- $V_1, ..., V_n$ are families of vectors in $\mathbb{R}^n$. Assume for all $S \subseteq [n]$ there are $|S|$ independent vectors $v_1^{(S)}, ..., v_{|S|}^{(S)}$ in $\cup_{i \in S} V_i$.

Rado's theorem with volume

A geometric corollary

- $V_1, ..., V_n$ are families of vectors in $\mathbb{R}^n$. Assume for all $S \subseteq [n]$ there are $|S|$ independent vectors $v_1^{(S)}, ..., v_{|S|}^{(S)}$ in $\cup_{i \in S} V_i$.
- R'3?: $\Rightarrow$ independent transversal $v_1 \in V_1, ..., v_n \in V_n$.

Rado's theorem with volume

A geometric corollary

- $V_1, \ldots, V_n$ are families of vectors in $\mathbb{R}^n$. Assume for all $S \subseteq [n]$ there are $|S|$ independent vectors $v_1^{(S)}, \ldots, v_{|S|}^{(S)}$ in $\cup_{i \in S} V_i$.
- R'3?: $\Rightarrow$ independent transversal $v_1 \in V_1, \ldots, v_n \in V_n$.
- Volume: Assume for all $S \subseteq [n]$ that $\mathrm{Vol}_{|S|}\left(\left[v_1^{(S)}, \ldots, v_{|S|}^{(S)}\right]\right) \geq \epsilon^{|S|}$.

Rado's theorem with volume

A geometric corollary

- $V_1, \ldots, V_n$ are families of vectors in $\mathbb{R}^n$. Assume for all $S \subseteq [n]$ there are $|S|$ independent vectors $v_1^{(S)}, \ldots, v_{|S|}^{(S)}$ in $\cup_{i \in S} V_i$.
- R'3?: $\Rightarrow$ independent transversal $v_1 \in V_1, \ldots, v_n \in V_n$.
- Volume: Assume for all $S \subseteq [n]$ that $\mathrm{Vol}_{|S|}\left(\left[v_1^{(S)}, \ldots, v_{|S|}^{(S)}\right]\right) \geq \epsilon^{|S|}$.
- $\Rightarrow$ an independent transversal $v_1 \in V_1, \ldots, v_n \in V_n$ with $\mathrm{Vol}_n([v_1, \ldots, v_n]) \geq \epsilon^{O(n^2)}$.

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds
  - $\sum_i B_i B_i^t = I$

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds
  - $\sum_i B_i B_i^t = I$
  - $\sum_i B_i^t B_i = I$

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds
  - $\sum_i B_i B_i^t = I$
  - $\sum_i B_i^t B_i = I$
- In this case even the existence of scaling factors has been a major open question.

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds
  - $\sum_i B_i B_i^t = I$
  - $\sum_i B_i^t B_i = I$
- In this case even the existence of scaling factors has been a major open question. The appropriate optimization problem is non-convex.

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds
  - $\sum_i B_i B_i^t = I$
  - $\sum_i B_i^t B_i = I$
- In this case even the existence of scaling factors has been a major open question. The appropriate optimization problem is non-convex.
- Was recently resolved by GGdOW'16 using geodesic convexity.

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds
  - $\sum_i B_i B_i^t = I$
  - $\sum_i B_i^t B_i = I$
- In this case even the existence of scaling factors has been a major open question. The appropriate optimization problem is non-convex.
- Was recently resolved by GGdOW'16 using geodesic convexity.
- Out of scope of this lecture.

Operator scaling

- Let $A_1, ..., A_m$ be m $n \times n$ matrices. Do there exist matrices (scaling factors) X and Y such that for $B_i = XA_iY$ holds
  - $\sum_i B_i B_i^t = I$
  - $\sum_i B_i^t B_i = I$
- In this case even the existence of scaling factors has been a major open question. The appropriate optimization problem is non-convex.
- Was recently resolved by GGdOW'16 using geodesic convexity.
- Out of scope of this lecture(r).

Thank you!