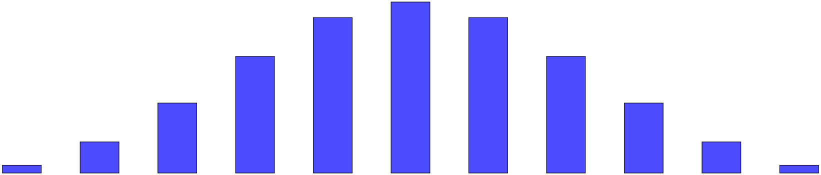
Complexity of Log-concave Inequalities for Matroids

Swee Hong Chan

joint with Igor Pak



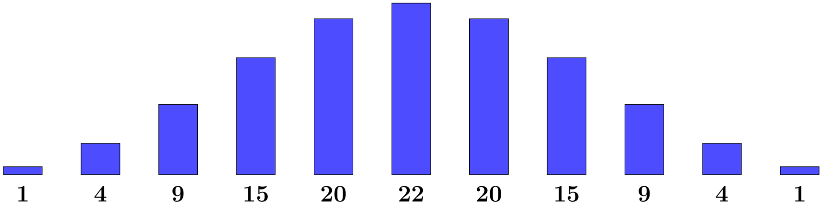
What is log-concavity?

A sequence $a_1, \ldots, a_n \in \mathbb{R}_{\geq 0}$ is log-concave if

$$a_k^2 \;\geq\; a_{k+1}\, a_{k-1} \qquad (1 < k < n).$$

Log-concavity (and positivity) implies unimodality:

$$a_1 \;\leq\; \cdots \;\leq\; a_m \;\geq\; \cdots \;\geq\; a_n \quad \text{for some } 1 \leq m \leq n.$$



Example 1: binomial coefficients

$$a_k = \binom{n}{k} \qquad k = 0, 1, \ldots, n.$$

This sequence is log-concave because

$$\frac{a_k^2}{a_{k+1}\, a_{k-1}} = \frac{\binom{n}{k}^2}{\binom{n}{k+1}\binom{n}{k-1}} = \left(1 + \frac{1}{k}\right)\left(1 + \frac{1}{n-k}\right),$$

which is greater than 1.

Example 2: permutations with k inversions

$$a_k \;=\; \text{number of } \pi \in S_n \text{ with } k \text{ inversions},$$

where inversion of π is pair $i < j$ s.t. $\pi_i > \pi_j$.

This sequence is log-concave because

$$\sum_{0 \le k \le \binom{n}{2}} a_k\, q^k \;=\; [n]_q! \;=\; \prod_{i=1}^{n-1} (1 + q + q^2 + \ldots + q^i)$$

is a product of log-concave polynomials.

Example 3: forests of a graph

a_k = number of forests with k edges of graph G.

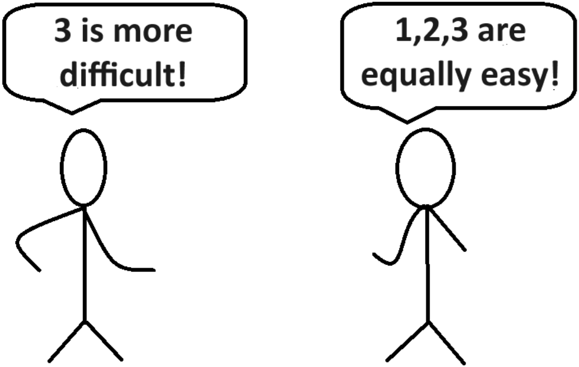
Forest is a subset of edges of G that has no cycles.

Log-concavity was conjectured for all matroids (Mason '72), and was proved through combinatorial Hodge theory (Huh '15).

G forest not forest spanning tree

Motivation

Which log-concave inequality is more "difficult"?



Our goal

We aim to differentiate simple log-concave inequalities from complex log-concave inequalities using **Complexity Theory**.

Matroids

Object: matroids

Matroid $\mathcal{M} = (X, \mathcal{I})$ is ground set X with collection of independent sets $\mathcal{I} \subseteq 2^X$.

Graphic matroids

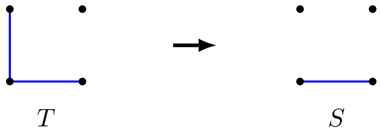
- X = edges of a graph G,
- $\mathcal{I}$ = forests in G.

Realizable matroids

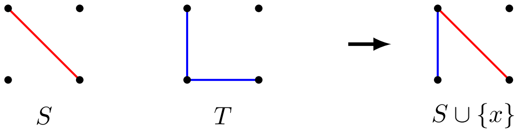
- X = finite set of vectors over field $\mathbb{F}$,
- $\mathcal{I}$ = sets of linearly independent vectors.

Matroids: conditions

- $S \subseteq T$ and $T \in \mathcal{I}$ implies $S \in \mathcal{I}$.



- If $S, T \in \mathcal{I}$ and $|S| < |T|$, then there is $x \in T \setminus S$ such that $S \cup \{x\} \in \mathcal{I}$.



A basis is a maximal independent set.

Rank r of matroid is the size of the bases.

Mason conjecture

For matroid $\mathcal{M}$, let

$\mathrm{I}(k) \;:=\;$ no. of independents sets with k elements.

For graphic matroid, $\mathrm{I}(k)$ is no. of forest with k edges.

Conjecture (Mason '72)

The sequence $\mathrm{I}(1), \mathrm{I}(2), \ldots$ *is log-concave,*

$$\mathrm{I}(k)^2 \;\geq\; \mathrm{I}(k+1)\,\mathrm{I}(k-1) \qquad (k \in \mathbb{N}),$$

Mason conjecture (continued)

Conjecture (Mason '72)

$$\mathrm{I}(k)^2 \;\geq\; \mathrm{I}(k+1)\,\mathrm{I}(k-1) \qquad (k \in \mathbb{N}).$$

Conjecture was proved for graphic matroids by (Huh '15), and for all matroids by (Adiprasito–Huh–Katz '18).

Both proofs used combinatorial Hodge theory.

Stanley–Yan inequality (simple)

Fix a disjoint subset S of X.

$\mathrm{B}(k) :=$ no. of bases B such that $|B \cap S| = k$,

multiplied by $r! \times \binom{r}{k}^{-1}$.

Theorem (Stanley '81, Yan '23)

The sequence $\mathrm{B}(1), \mathrm{B}(2), \ldots$ *is log-concave,*

$$\mathrm{B}(k)^2 \geq \mathrm{B}(k+1)\,\mathrm{B}(k-1) \qquad (k \in \mathbb{N}).$$

Stanley–Yan inequality (simple)

Theorem (Stanley '81, Yan '23)

$$\mathrm{B}(k)^2 \ \geq\ \mathrm{B}(k+1)\,\mathrm{B}(k-1) \qquad (k \in \mathbb{N}).$$

Proved for regular matroids by (Stanley '81) using Alexandrov–Fenchel inequality for mixed volumes.

Proved for all matroids by (Yan '23) using theory of Lorentzian polynomials.

Proof of Mason conjecture

Let

$$\mathcal{M}' := \text{direct sum of } \mathcal{M} \text{ with the free matroid of } r \text{ elements;}$$

$$S := X.$$

Then

$$\mathrm{I}(k) \text{ for } \mathcal{M} \quad = \quad \frac{1}{r!} \times \mathrm{B}(k) \text{ for } \mathcal{M}'.$$

Thus Stanley–Yan inequality for $\mathcal{M}'$ implies Mason conjecture for $\mathcal{M}$.

Stanley–Yan inequality (true)

Fix $d \geq 0$, disjoint subsets $S, S_1, \ldots, S_d$ of X, and $\ell_1, \ldots, \ell_d \in \mathbb{N}$.

$\mathrm{B}_d(k)$:= number of bases B of $\mathcal{M}$ such that $|B \cap S| = k$, $|B \cap S_i| = \ell_i$ for $i \in [d]$,

multiplied by $r! \times \binom{r}{k, \ell_1, \ldots, \ell_d}^{-1}$.

Theorem (Stanley '81, Yan '23)

The sequence $\mathrm{B}_d(1), \mathrm{B}_d(2), \ldots$ *is log-concave,*

$$\mathrm{B}_d(k)^2 \;\geq\; \mathrm{B}_d(k+1)\,\mathrm{B}_d(k-1) \qquad (k \in \mathbb{N}).$$

When is equality achieved?

Quote (Gardner '02)

If inequalities are silver currency in mathematics, those that come along with precise equality conditions are gold.

Example (AM–GM inequality)

For non-negative $x_1, \ldots, x_n$,

$$\frac{x_1 + \cdots + x_n}{n} \geq \sqrt[n]{x_1 \cdots x_n},$$

with equality if and only if $x_1 = \cdots = x_n$.

Main result

Consider the decision problem for checking equality in Stanley–Yan inequality:

$$\mathrm{B}_d(k)^2 \overset{?}{=} \mathrm{B}_d(k+1)\,\mathrm{B}_d(k-1).$$

Theorem (C.–Pak '24+)

For $d \geq 1$, problem cannot be decided in polynomial time, unless $\mathrm{NP} = \mathrm{coNP}$.

Main result

Consider the decision problem for checking equality in Stanley–Yan inequality:

$$\mathrm{B}_d(k)^2 \overset{?}{=} \mathrm{B}_d(k+1)\,\mathrm{B}_d(k-1).$$

Theorem (C.–Pak '24+)

For $d \geq 1$, problem cannot be decided in polynomial time, unless NP = coNP.

Theorem 1 (C.–Pak '24+)

For $d \geq 1$, problem is not part of **polynomial hierarchy**, *unless polynomial hierarchy collapses.*

Polynomial hierarchy

Decision vs counting

Decision problem: answer is either '`Yes`' or '`No`'.

Counting problem: answer is a nonnegative integer.

Example (3-colorings of graph G)

- *Decision problem: Check if there exists a proper 3-coloring of G.*
- *Counting problem: Find the number of proper 3-colorings of G.*

Polynomial hierarchy is a subclass of decision problems.

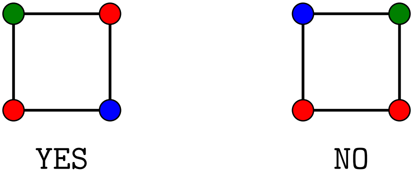
Complexity class P

$$\mathrm{P} := \left\{ \begin{array}{l} \text{Decision problems solvable by deterministic} \\ \text{Turing machine in polynomial time} \end{array} \right\}$$

Example

Check if a given 3-coloring of a graph G is proper.

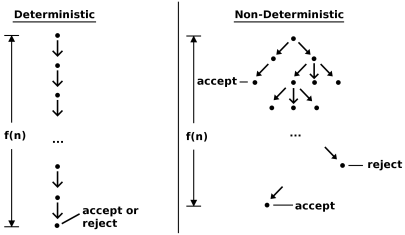
This can be solved in $O(n^2)$ time by checking the color of endpoints of every edge.



Complexity class NP

$$\mathrm{NP} := \left\{ \begin{array}{l} \text{Decision problems solvable by nondetermi-} \\ \text{nistic Turing machine in polynomial time} \end{array} \right\}.$$

- Can split into many parallel branches;
- Output '`YES`' if one of the branches said '`YES`';
- Output '`NO`' if all branches said '`NO`'.



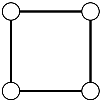
Complexity class NP

$$\text{NP} := \left\{ \begin{array}{l} \text{Decision problems solvable by nondetermi-} \\ \text{nistic Turing machine in polynomial time} \end{array} \right\}.$$

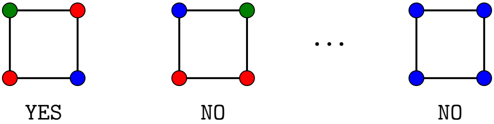
- Can split into many parallel branches;
- Output ‘YES’ if one of the branches said ‘YES’;
- Output ‘NO’ if all branches said ‘NO’.

Complexity class NP: example

Problem: Check if graph G has a proper 3-coloring.



Each branch corresponds to checking if a particular 3-coloring of G is proper.

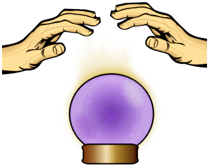


Output to this example is '`YES`'.

Turing machine with an oracle

At each step, this machine can either:

- Perform usual nondeterministic Turing machine operation; or
- Ask an oracle that is able to answer one fixed type of problems.



Turing machine with an oracle: example

Problem: Check if there is an induced subgraph of G of size $\lceil n/2 \rceil$ that is not 3-colorable.

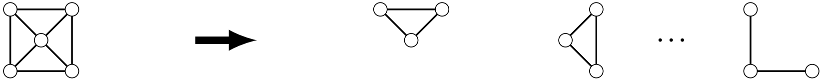
Oracle: Can check if a graph is 3-colorable.

Turing machine with an oracle: example

Problem: Check if there is an induced subgraph of G of size $\lceil n/2 \rceil$ that is not 3-colorable.

Oracle: Can check if a graph is 3-colorable.

Each branch of the machine corresponds to an induced subgraph of G of size $\lceil n/2 \rceil$.



For every branch, oracle checks if subgraph is 3-colorable.

Complexity class Σ_i^{P}

The first two classes are

$$\Sigma_0^{\mathsf{P}} := \mathsf{P}; \qquad \Sigma_1^{\mathsf{P}} := \mathsf{NP}.$$

For $i \geq 1$, the class $\Sigma_i^{\mathsf{P}} := \mathsf{NP}^{\Sigma_{i-1}^{\mathsf{P}}}$ is

$$\left\{ \begin{array}{l} \text{Decision problems solvable by nondetermi-} \\ \text{nistic Turing machine in polynomial time} \\ \text{with an oracle for problem from } \Sigma_{i-1}^{\mathsf{P}}. \end{array} \right\}.$$

Note that

$$\Sigma_0^{\mathsf{P}} \subseteq \Sigma_1^{\mathsf{P}} \subseteq \Sigma_2^{\mathsf{P}} \subseteq \Sigma_3^{\mathsf{P}} \subseteq \cdots$$

Polynomial hierarchy (PH)

Polynomial hierarchy is the union of all Σ_i^{P}'s,

$$\mathsf{PH} := \bigcup_{i=0}^{\infty} \Sigma_i^{\mathsf{P}}.$$

Conjecture

Polynomial hierarchy does not collapse,

$$\Sigma_0^{\mathsf{P}} \subsetneq \Sigma_1^{\mathsf{P}} \subsetneq \Sigma_2^{\mathsf{P}} \subsetneq \Sigma_3^{\mathsf{P}} \subsetneq \cdots$$

- $\Sigma_0^{\mathsf{P}} = \Sigma_1^{\mathsf{P}}$ is equivalent to $\mathsf{P} = \mathsf{NP}$.
- $\Sigma_1^{\mathsf{P}} = \Sigma_2^{\mathsf{P}}$ is equivalent to $\mathsf{NP} = \mathsf{coNP}$.

Back to main result

Consider the decision problem for checking equality in Stanley–Yan inequality:

$$\mathrm{B}_d(k)^2 \overset{?}{=} \mathrm{B}_d(k+1)\,\mathrm{B}_d(k-1).$$

Theorem (C.–Pak '24+)

For $d \geq 1$, decision problem is not in **PH**, *unless* PH *collapses.*

Recall our goal ...

We aim to differentiate simple log-concave inequalities from complex log-concave inequalities using **Complexity Theory**.

Complexity class $\#P$

Complexity class $\#P$

$$\#\text{P} := \left\{ \begin{array}{l} \text{Counting problems realizable as number} \\ \text{of `YES' branches in some nondetermi-} \\ \text{nistic Turing machine.} \end{array} \right\}.$$

Example

Count number of proper 3-colorings of graph G.

Not to be confused with FP, which is counting problems solvable in deterministic polynomial time.

Main result

Theorem 2 (C.–Pak '24+)

For $d \geq 1$, the defect of Stanley–Yan inequality

$$\mathrm{B}_d(k)^2 \;-\; \mathrm{B}_d(k+1)\,\mathrm{B}_d(k-1)$$

is not in $\#\mathrm{P}$, unless PH *collapses.*

Note: $\mathrm{B}_d(k)^2$ and $\mathrm{B}_d(k+1)\,\mathrm{B}_d(k-1)$ are in $\#P$.

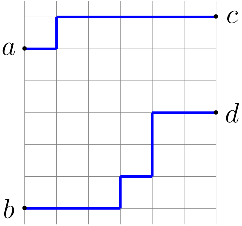
Example 1: binomial inequality

$$\binom{n}{k}^2 \geq \binom{n}{k+1}\binom{n}{k-1} \qquad (1 < k < n).$$

This inequality has a lattice path interpretation:

$K(a \to c, b \to d) :=$ no. of pairs of north-east lattice paths from a to c and b to d,

for $a, b, c, d \in \mathbb{Z}^2$.



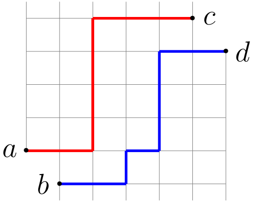
Example 1: binomial inequality

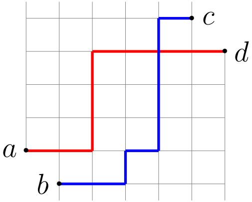
Let

$$a = (0,1), \qquad c = (k, n-k+1),$$
$$b = (1,0), \qquad d = (k+1, n-k).$$

Then

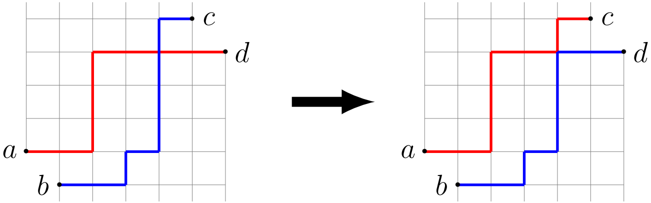
$$K(a \to c, b \to d) = \binom{n}{k}^2,$$
$$K(a \to d, b \to c) = \binom{n}{k-1}\binom{n}{k+1}.$$





Example 1: binomial inequality

Note $K(a \to c, b \to d) \geq K(a \to d, b \to c)$ by path-swapping injections.



$K(a \to c, b \to d) - K(a \to d, b \to c)$ is number of pairs of north-east lattice paths from a to c, b to d, that do not intersect.

Thus this number is in #P.

Example 2: permutations with k inversions

Let a_k = number of $\pi \in S_n$ with k inversions.

Then $$\sum_{0 \le k \le \binom{n}{2}} a_k \, q^k \; = \; \prod_{i=1}^{n-1} (1 + q + \ldots + q^i)$$

is computable in poly(n) time.

Thus $a_k^2 - a_{k+1} a_{k-1}$ is computable in poly(n) time,

and so is in #P.

Back to our goal

We compare three log-concave inequalities:

Binomial inequality: in #P;

Permutation inversion inequality: in #P;

Stanley–Yan inequality: not in #P, unless PH collapses.

This differentiates Stanley–Yan inequality from binomial inequality and permutation inversion inequality.

What is next?

Conjecture

Defect of Mason conjecture

$$\mathrm{I}(k)^2 \; - \; \mathrm{I}(k+1)\,\mathrm{I}(k-1) \quad \notin \; \#\mathsf{P}.$$

THANK YOU!

Preprint: www.arxiv.org/abs/2309.05764

Webpage: www.math.rutgers.edu/~sc2518/

Email: sweehong.chan@rutgers.edu

Complexity class Σ_i^{P}: example

Problem A: Check if a 3-coloring of G is proper.
Problem A is in $\Sigma_0^{\mathsf{P}} = \mathsf{P}$.

Problem B: Check if G has a proper 3-coloring.
Problem B is in $\Sigma_1^{\mathsf{P}} = \mathsf{NP}$.

Problem C: Check if there is an induced subgraph of G of size $\lceil n/2 \rceil$ that is not 3-colorable.
Problem C is in $\Sigma_2^{\mathsf{P}} = \mathsf{NP}^{\mathsf{NP}}$.