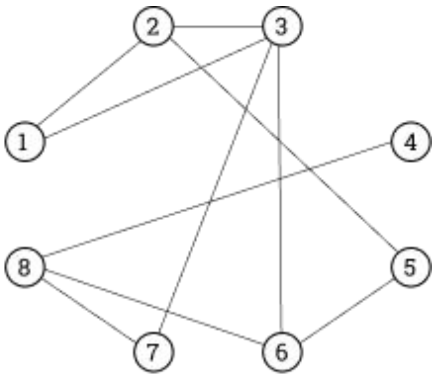
Sum-of-Squares Lower Bounds for Sparse Independent Set

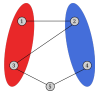


Chris Jones (University of Chicago) ← on the postdoc market
Aaron Potechin (University of Chicago)
Goutham Rajendran (University of Chicago)
Madhur Tulsiani (Toyota Technological Institute at Chicago)
Jeff Xu (Carnegie Mellon University)

Outline

🆘 The Sum-of-Squares algorithm and Independent Set

😢 Pseudocalibration doesn't work

 Matrix-valued Fourier analysis

🔚 Conclusion

The Sum-of-Squares algorithm and Independent Set

The Sum-of-Squares (SoS) Algorithm

SoS is a super strong semidefinite programming (SDP) algorithm

- SDP = linear program over the set of positive semidefinite (PSD) matrices

SoS can be used to optimize a polynomial subject to polynomial constraints

SoS consists of complete local reasoning (=Sherali-Adams) and PSD-ness

SoS captures our best algorithms for Max Cut [GW'95], Sparsest Cut [ARV'04], Unique Games [BBKSS'21], Tensor PCA [HSS'15], robust linear regression [BP'21], ...

- PSD-ness is critically important for all of these applications
- PSD-ness is also what makes proving lower bounds against SoS so hard

[Goemans-WIlliamson '95] [Arora-Rao-Vazirani '04]
[Bafna-Barak-Kothari-Schramm-Steurer '21] [Hopkins-Shi-Steurer '15]
[Bakshi-Prasad '21]

The Sum-of-Squares (SoS) Algorithm

SoS_D is actually a hierarchy of convex relaxations, one for each degree D.
As D gets larger, the convex program gets larger and larger (slower runtime), but the value of SoS_D approaches the true value.

$$\mathrm{val}(SoS_2) \geq \mathrm{val}(SoS_4) \geq \mathrm{val}(SoS_6) \geq \dots \geq \mathrm{val}(SoS_{2n-2}) \geq \mathrm{val}(SoS_{2n}) = \alpha(G)$$

Lemma. SoS_D can be expressed as a semidefinite program of size $n^{O(D)}$

D = O(1) corresponds to polynomial time
D = 2n solves the problem exactly in exponential time (for n Boolean variables)

Main Question: how large must D be before $\mathrm{val}(SoS_D) \approx \alpha(G)$?

How to prove an SoS lower bound

Main Question: how large must D be before val(SoS_D) ≈ α(G)?

To prove a lower bound, prove that val(SoS_D) ≫ α(G) for D = some ω(1)

□This is certified by a dual object, which is a degree-D pseudoexpectation operator **$\tilde{\mathbf{E}}$**

Previous techniques for constructing and analyzing **$\tilde{\mathbf{E}}$**:

- Exact PSD factorization for 3XOR [Grigoriev'99, Schoenebeck'08]
- Symmetric problems [CSS'14, Potechin'18]
- Gram-Schmidt for random, "pairwise uniform" CSPs [KMOW'17]
- Approximate PSD factorization for random, dense inputs [BHKKMP'16, PR'20, GJJPR'20]
- Lift degree-2 to higher degree [MRX'19, Kunisky'20]

Our contribution: develop lower bound techniques for random, sparse inputs

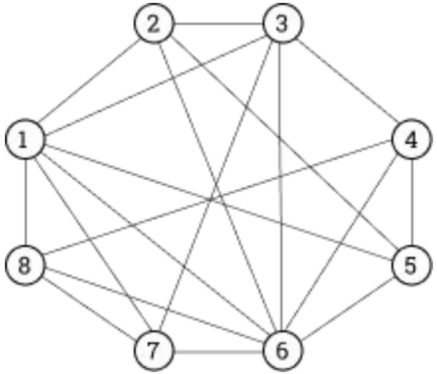
1. *The Sum-of-Squares algorithm and independent set*

[Codenotti-Schoenebeck-Snook '14] [Kothari-Mori-O'Donnell-Witmer '17]
[Barak-Hopkins-Kelner-Kothari-Moitra-Potechin '16] [Potechin-Rajendran '20]
[Ghosh-Jeronimo-Jones-Potechin-Rajendran '20]
[Mohanty-Raghavendra-Xu '19]

Dense vs Sparse

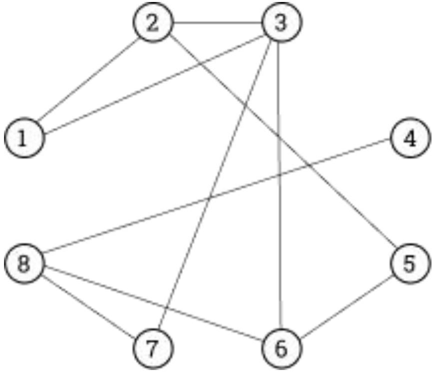
Dense: uniform $\{-1,+1\}^n$ or G(n,½) or a collection of Gaussians

- More generally, the input random variables are O(1)-subgaussian



Sparse: G(n,p) or random d-regular graph (p=d/n, d = o(n))

- Moments/probability distributions of inputs depend on n
- Our techniques work best for d ∈ [polylog(n), o(n)]



Independent Set

Given an n-vertex graph G, what is $\alpha(G)$ = largest independent set in G?

In a random graph $G \sim G(n, d/n)$ with $d = o(n)$, with high probability...

Theorem [Frieze'90]. $\alpha(G) = (2+o_d(1))\, n \ln(d) / d$

Theorem [Coja-Oghlan'05]. $\vartheta(G) = \mathrm{val}(SoS_2) = d^{1/2-o(1)}\ \alpha(G)$

Theorem [this talk]. Assuming $d \gg \log(n)$, for $D = n^{o(1)}$, $\mathrm{val}(SoS_D) = d^{1/2-o(1)}\,\alpha(G)$.

That is, there is a degree-D $\tilde{E}$ satisfying "G has an ind set of size $d^{1/2-o(1)}\,\alpha(G)$"

Pseudocalibration doesn't work

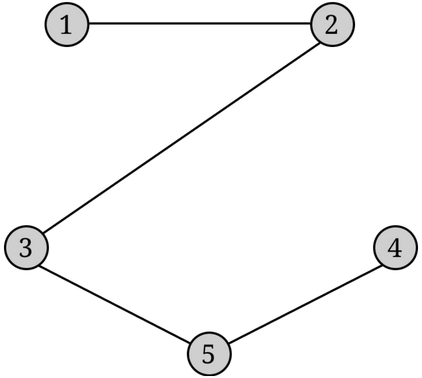
Boolean Fourier analysis

Recall: Boolean function $f : \{-1,+1\}^n \to \mathbb{R}$ can be represented in the Fourier basis

- This is just its representation as a multilinear polynomial
- There is one monomial/Fourier character χ_S for each subset $S \subseteq [n]$
- Useful when f is a function of a *random* bitstring x, because the basis is orthonormal

In the Independent Set problem, the input is a graph $G \in \{-1,+1\}^{\binom{n}{2}}$

- There is one monomial/Fourier character χ_H for each graph H on [n]



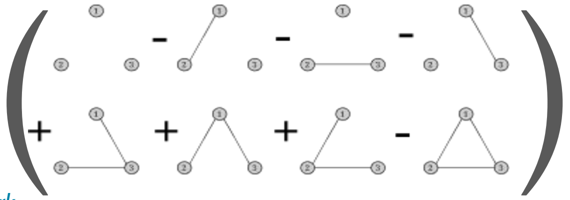
Example:

$\chi_H = G_{1,2}G_{2,3}G_{3,5}G_{4,5}$

Example:

{1,2,3} are a triangle =

$(1 - G_{1,2})(1-G_{1,3})(1-G_{2,3})/8$



/ 8

Pseudocalibration

We want to construct $\tilde{\mathbb{E}}$ that "thinks" there is a large independent set

$\tilde{\mathbb{E}}$ is a *fake* expectation over independent sets, specified by moments up to degree D

$\tilde{\mathbb{E}}[X^V]$ for each $V \subseteq [n]$, $|V| \leq D$, $\tilde{\mathbb{E}}[X_1] = 0.2$, $\tilde{\mathbb{E}}[X_7 X_{10}] = 0.75$

[BHKKMP'16] suggests pseudocalibrating $\tilde{\mathbb{E}}$ using a planted distribution D

Suppose (G', S) ~ D is a distribution on graphs G' with large ind sets
When given the (non-planted) input graph G ~ G(n,p), for $V \subseteq [n]$ define:

$$\tilde{\mathbb{E}}[X^V] = \sum_{H : |V \cup V(H)| \leq \tau} \mathop{\mathbb{E}}_{(G',S)\sim D} [S^V \cdot \chi_H(G')] \cdot \chi_H(G)$$

[Barak-Hopkins-Kelner-Kothari-Moitra-Potechin '16]

Pseudocalibration

Natural planted distribution for an Independent Set of size k:

- Sample G ~ G(n,p)
- Plant an independent set of expected size k: pick each vertex independently with probability k/n and remove edges between picked vertices

Problem: this distribution is distinguishable from G(n,p) with probability Ω(1)

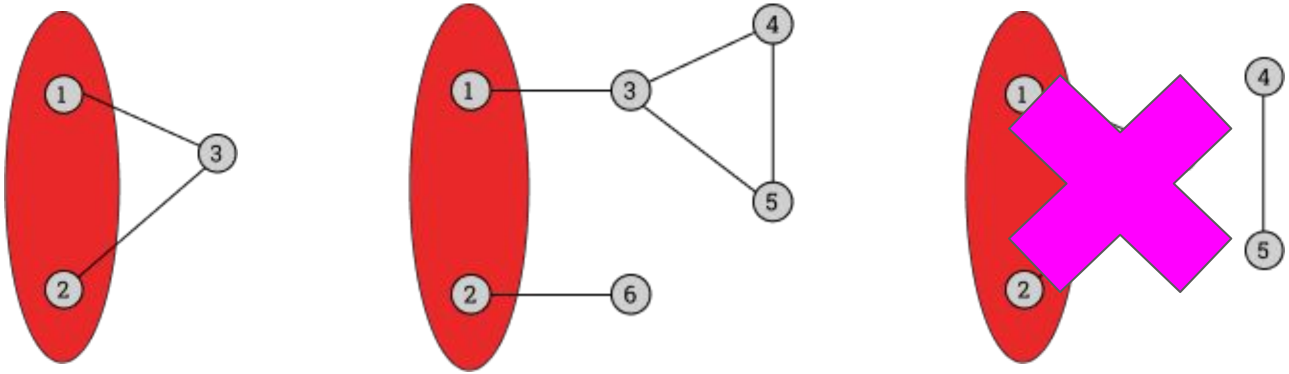
- Distinguisher: count the number of edges in G
- In all previous successful uses of pseudocalibration, the planted and random distributions are (conjecturally) hard to distinguish by all polynomial time algorithms

Connected Truncation

Solution: truncate away the global tests when defining **Ẽ**

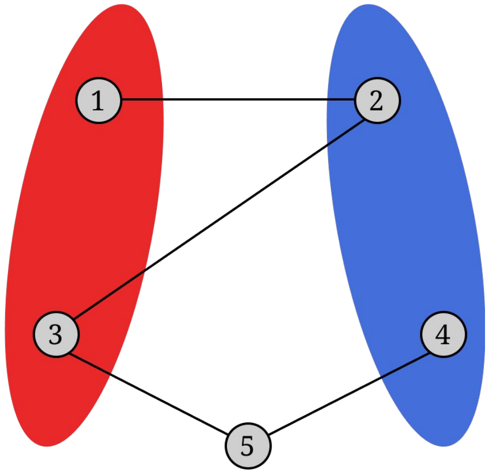
$$\tilde{E}[X^V] = \sum_{\substack{H:|V \cup V(H)| \le \tau, \\ H \text{ connected to } V}} \mathop{\mathbb{E}}_{(G',S)\sim D} [S^V \cdot \chi_H(G')] \cdot \chi_H(G)$$

V = {1, 2}



All that remains is to prove that **Ẽ** is PSD

Matrix-valued Fourier analysis

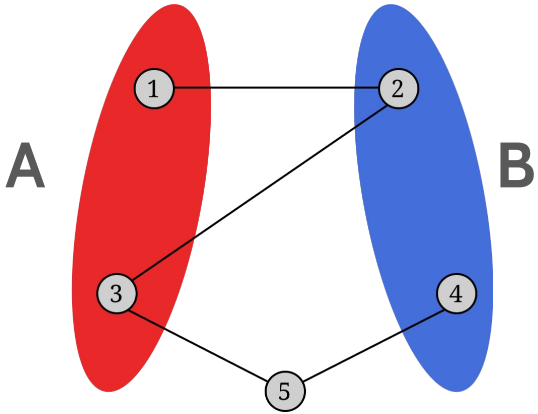


Boolean Fourier analysis: matrix edition

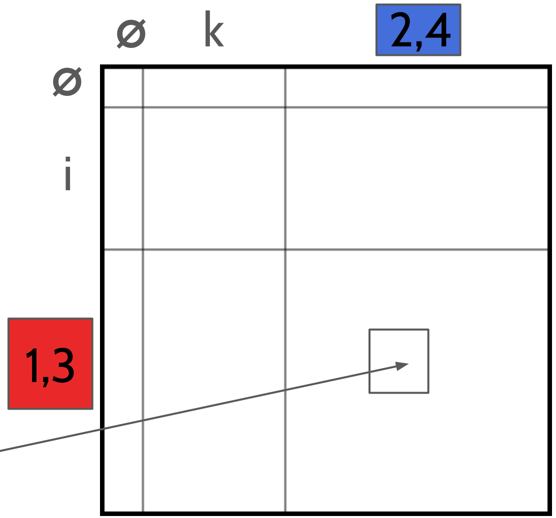
We need to analyze matrix-valued functions of the graph G

- Each entry of the matrix is itself a function of G

Def: a ribbon is a particular Fourier character in a particular matrix entry. It is specified by a graph H on [n] and two index sets A, B ⊆ [n]

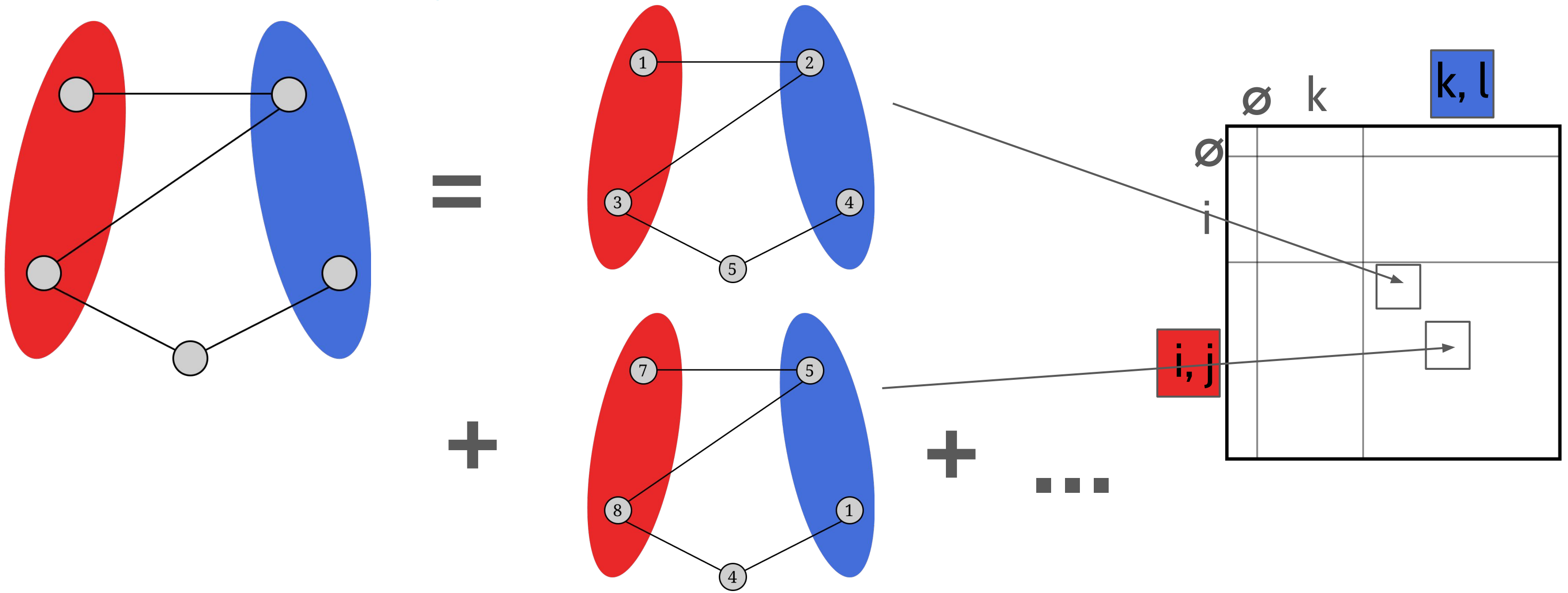


$G_{1,2}G_{2,3}G_{3,5}G_{4,5}$



Boolean Fourier analysis: matrix edition

Def: graph matrix M_α sums together all ribbons with a particular shape α



Theorem. Where max is over vertex separators S of the left and right sides of α, whp:

$$\|M_\alpha\| \le \max_S \tilde{O}\left(n^{|V(\alpha)|/2 + \mathrm{Isolated}(\alpha)/2}\, n^{-|V(S)|/2} p^{-|E(S)|/2}\right)$$

Boolean Fourier analysis: matrix edition

The spectral norm of a graph matrix is determined by combinatorial properties of the shape!

Dense setting

Theorem [AMP'16].

$$\|M_\alpha\| \le \max_S \tilde{O}\left(n^{|V(\alpha)|/2 + \text{Isolated}(\alpha)/2}\, n^{-|V(S)|/2}\right)$$

Sparse setting

Theorem [this talk].

$$\|M_\alpha\| \le \max_S \tilde{O}\left(n^{|V(\alpha)|/2 + \text{Isolated}(\alpha)/2}\, n^{-|V(S)|/2} p^{-|E(S)|/2}\right)$$

We use an approximate PSD decomposition [BHKKMP'16].

Vertex factors can be handled in a similar way to [BHKKMP'16].

For edge factors, we give new charging arguments.

[Ahn-Medarametla-Potechin '16]
[Barak-Hopkins-Kelner-Kothari-Moitra-Potechin '16]

Further technical details

- Spectral norms are not concentrated enough due to rare events (presence of small dense subgraphs, e.g. K_8). Condition on no such subgraphs.
- Some spectral norms should be controlled using the Frobenius norm
- Matrix for $\tilde{\mathbf{E}}$ has a nullspace. Factor it out with missing edge indicators.
- Use quasi-missing edge indicators and missing edge indicators to handle edges incident to $U_\alpha \cap V_\alpha$
- Even with all this work, there is still one log factor in our bound. Removing it would allow us to prove bounds for the regime of constant d

🔚 Conclusion

We develop lower bound technology for SoS in the sparse setting

- Even though there isn't a planted distribution, we amend pseudocalibration by using a connected truncation
- Prove norm bounds for sparse graph matrices
- Combinatorial analysis using conditioning and apx PSD factorization

Open problems:

- Is there a planted distribution which is hard to distinguish from G(n,p)?
- Extend Independent Set lower bound to the regime of constant d
- SoS lower bounds for other sparse problems: MaxCut on a random degree-d graph, Densest-k-Subgraph

🥳Thanks for watching!🥳