Computational Phase Transitions in Tensor Decomposition

Alex Wein

University of California, Davis

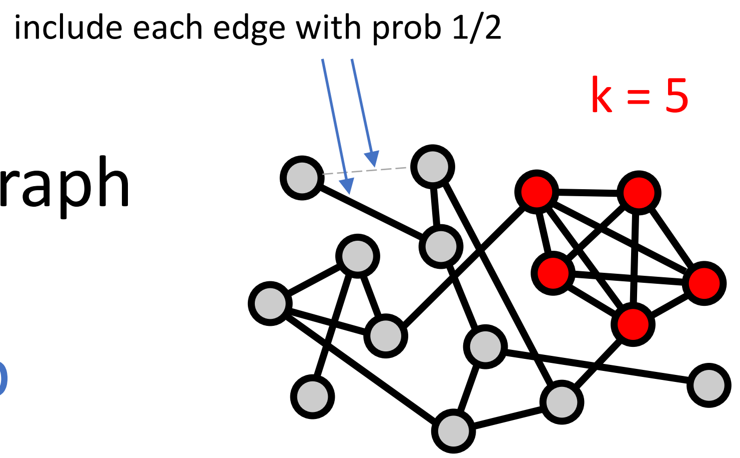
Department of Mathematics

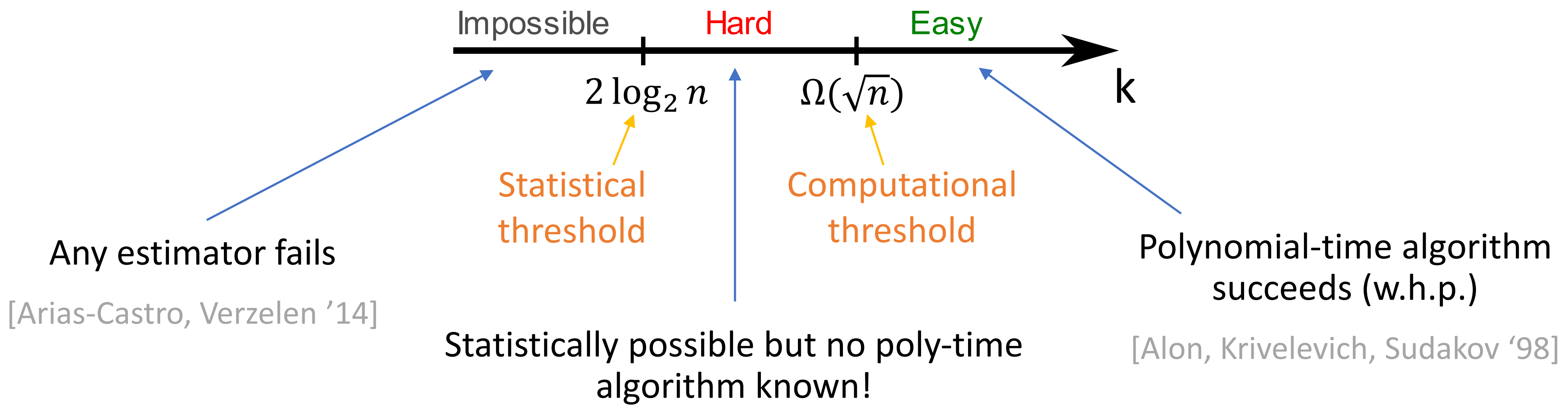
UCDAVIS

Part 1
Intro to statistical-computational gaps and the low-degree polynomial framework

Example: Planted Clique

- Find a planted k-clique in an n-vertex random graph
  - G(n,1/2) + {random k-clique}
- Believed to have a statistical-computational gap





Not Just Planted Clique...

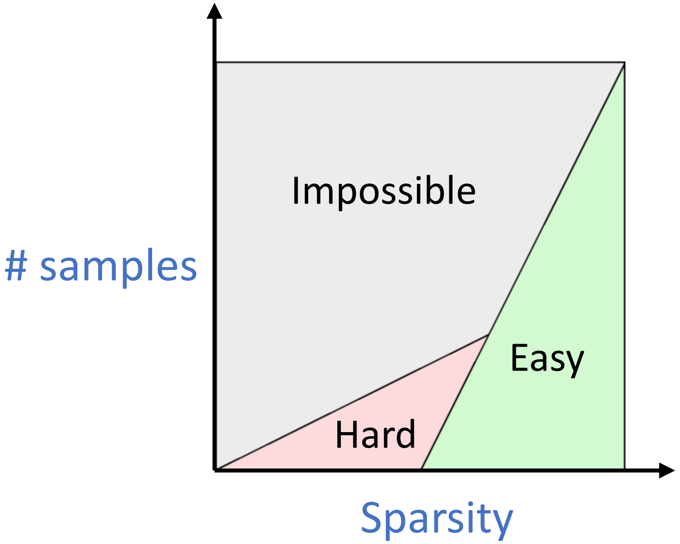
- Sparse PCA
- Community detection (SBM)
- Tensor PCA
- Random CSPs
- Spiked Wigner model
- Spiked Wishart model
- Planted submatrix
- Planted dense subgraph
- Planted vector in a subspace
- Dictionary learning

- Non-gaussian component analysis
- Independent component analysis
- Tensor decomposition
- Sparse linear regression
- Phase retrieval
- Group testing
- Generalized linear models
- Synchronization
- Orbit recovery

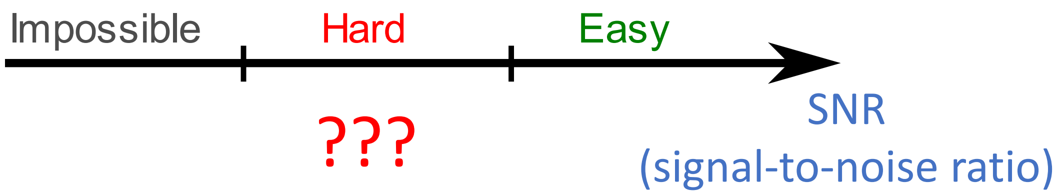
- Gaussian clustering
- Sparse clustering
- Matrix completion
- Tensor completion
- Graph matching
- Planted matching
- Mixed membership SBM
- Hypergraphic planted clique
- Secret leakage planted clique
- Continuous learning with errors

- Robust sparse mean estimation
- Certifying RIP
- Spiked transport model
- Hidden hubs
- Planted coloring
- Number partitioning
- Nonnegative PCA
- Cone-constrained PCA
- Sparse tensor PCA
- Robust sparse PCA

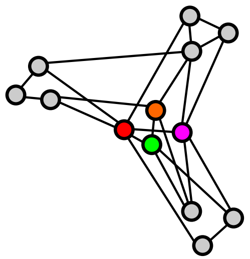
- Learning neural networks
- Sherrington-Kirkpatrick model
- Spin glass optimization
- ...



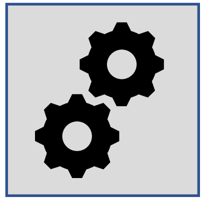
Statistical-computational gaps are ubiquitous!



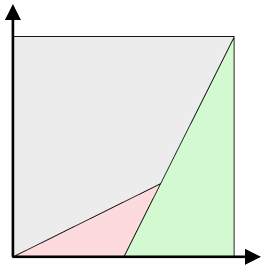
The Dream



New statistical problem



Systematic method
of analysis



Phase diagram

"Impossible" phase: classical statistics (Assouad, Fano, Le Cam, …)

"Easy" phase: algorithm design (spectral methods, message passing, …)

"Hard" phase: need evidence for computational hardness

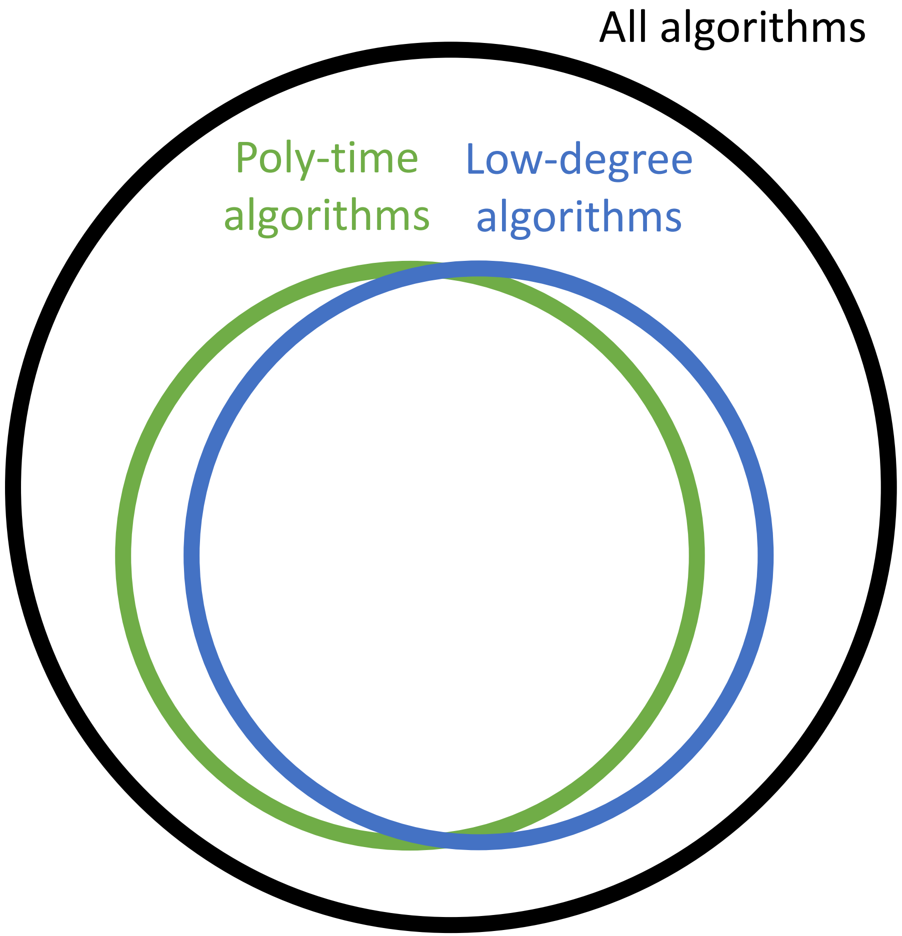
~~NP-hardness~~ "computational complexity of statistical inference"

A Restricted Class of Algorithms

Low-degree polynomial algorithms

- ☑ Good "proxy" for poly-time algorithms, for statistical problems
- ☑ Tractable to analyze
- ☑ Widely applicable
- ☑ Unified explanation for hardness

Other restricted classes: sum-of-squares, statistical query model, AMP, …



Low-Degree Polynomial Algorithms

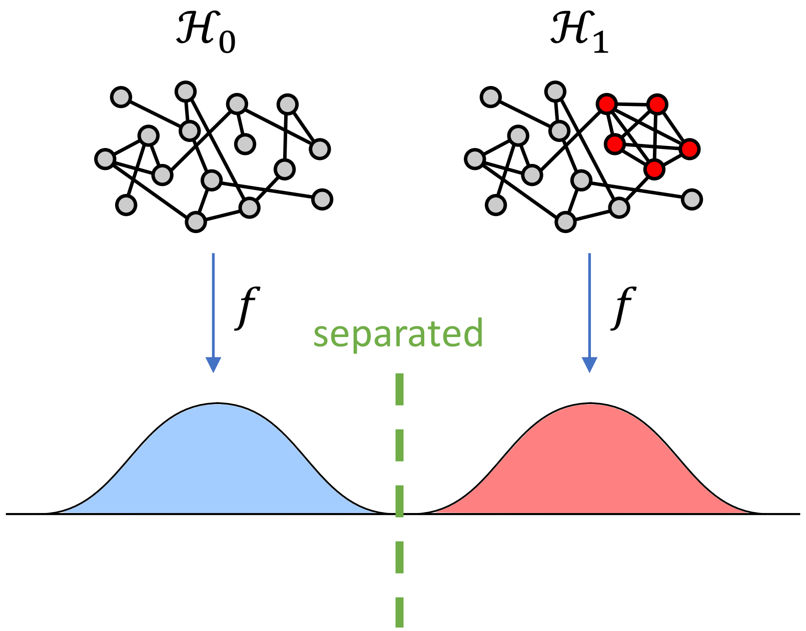
[Hopkins, Steurer '17; Hopkins, Kothari, Potechin, Raghavendra, Schramm, Steurer '17; Hopkins '18; Kunisky, **W**, Bandeira '19]

- **Hypothesis testing**: "is there a planted signal?"
  - Distinguish $\mathcal{H}_0$ (random graph) vs $\mathcal{H}_1$ (planted clique)
  - Goal: vanishing error probability
- **Low-degree polynomial algorithm**: multivariate polynomial of degree O(log n)

$$f: \{0,1\}^{\binom{n}{2}} \to \mathbb{R}$$

Input: graph

Output: number



"Success": f's output is "small" under $\mathcal{H}_0$, "large" under $\mathcal{H}_1$

$$\sqrt{\max\{\mathrm{Var}_{\mathcal{H}_0}(f), \mathrm{Var}_{\mathcal{H}_1}(f)\}} = o\left(\mathrm{E}_{\mathcal{H}_1}[f] - \mathrm{E}_{\mathcal{H}_0}[f]\right)$$

Low-Degree Polynomial Algorithms

[Hopkins, Steurer '17; Hopkins, Kothari, Potechin, Raghavendra, Schramm, Steurer '17; Hopkins '18; Kunisky, **W**, Bandeira '19]

- **Hypothesis testing**: "is there a planted signal?"
  - Distinguish $\mathcal{H}_0$ (random graph) vs $\mathcal{H}_1$ (planted clique)
  - Goal: vanishing error probability
- **Low-degree polynomial algorithm**: multivariate polynomial of degree O(log n)

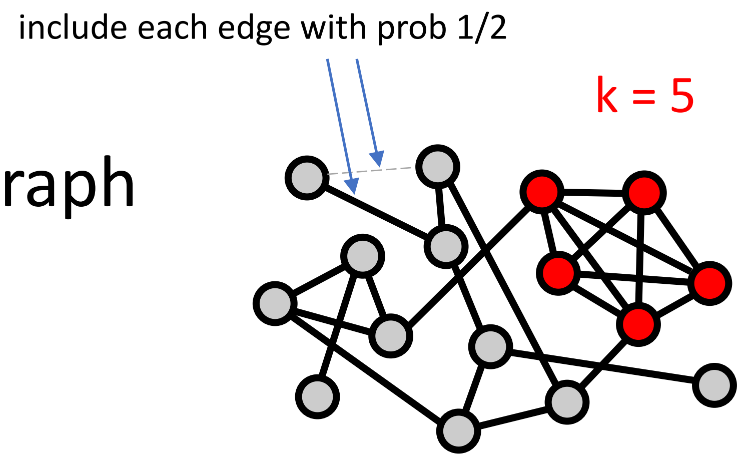
$$f: \{0,1\}^{\binom{n}{2}} \to \mathbb{R}$$

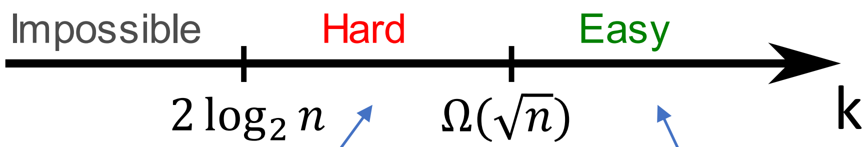
Input: graph — Output: number

- **Examples**:
  - Count total edges: $f(A) = \sum_{i<j} A_{ij}$
  - Count triangles: $f(A) = \sum_{i<j<k} A_{ij} A_{ik} A_{jk}$
  - Leading eigenvalue of adjacency matrix: $\lambda_{\max}^{2p} \approx \sum_i \lambda_i^{2p} = \mathrm{Tr}(A^{2p})$ $\quad p \approx \log n$

Back to Planted Clique

- ~~Find~~ Detect a planted k-clique in an n-vertex random graph
  - G(n,1/2) + {random k-clique}





When $k \ll \sqrt{n}$, **all** low-degree algorithms fail
[Barak, Hopkins, Kelner, Kothari, Moitra, Potechin ’16; Hopkins ‘18]

When $k \gg \sqrt{n}$, some low-degree algorithm succeeds
(degree-1 polynomial: count total edges)

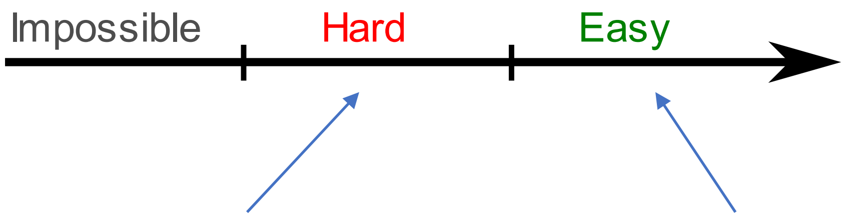
Low-Degree Algorithms Are Optimal??

Stellar track record for capturing the computational threshold

Planted clique [Barak, Hopkins, Kelner, Kothari, Moitra, Potechin '16; Hopkins '18]
Community detection [Hopkins, Steurer '17; Hopkins '18]
Mixed membership SBM [Hopkins, Steurer '17]
Tensor PCA [Hopkins, Kothari, Potechin, Raghavendra, Schramm, Steurer '17]
Spiked Wishart [Bandeira, Kunisky, **W** '20]
Spiked Wigner [Kunisky, **W**, Bandeira '19]
Sparse PCA [Ding, Kunisky, **W**, Bandeira '19]
Max independent set [Gamarnik, Jagannath, **W** '20; **W** '20]
Secret leakage planted clique [Brennan, Bresler '20]
Hypergraphic planted clique [Luo, Zhang '20]

Sparse clustering [Löffler, **W**, Bandeira '20]
Certifying RIP [Ding, Kunisky, **W**, Bandeira '21]
Planted submatrix [Schramm, **W** '20]
Planted dense subgraph [Schramm, **W** '20]
Multi-spiked Wigner/Wishart [Bandeira, Banks, Kunisky, Moore, **W** '21]
Planted affine planes [Ghosh, Jeronimo, Jones, Potechin, Rajendran '20]
Gaussian mixture models [Brennan, Bresler, Hopkins, Li, Schramm '21]
Gaussian graphical models [Brennan, Bresler, Hopkins, Li, Schramm '21]
Morris class of exponential families [Kunisky '20]
Robust sparse PCA [d'Orsi, Kothari, Novikov, Steurer '20]

Non-negative PCA [Bandeira, Kunisky, **W** '21]
Planted vector in a subspace [Mao, **W** '21]
Random k-SAT [Bresler, Huang '21]
Sparse tensor PCA [Choo, d'Orsi '21]
Refuting random polynomial systems [Hsieh, Kothari '21]
Sparse linear regression [Arpino '21; Bandeira, Alaoui, Hopkins, Schramm, **W**, Zadik '22]
Gaussian clustering [Mao, **W** '21, Davis, Diaz, Wang '21]
Graph matching [Mao, Wu, Xu, Yu '21]
Group testing [Coja-Oghlan, Gebhard, Hahn-Klimroth, **W**, Zadik '22]



No known poly-time alg
All low-deg algs provably fail

Some poly-time alg provably succeeds
Some low-deg alg provably succeeds

"Low-Degree Conjecture" [Hopkins '18]: low-degree algorithms are optimal among all polynomial-time algorithms for *natural high-dimensional statistical problems**.

Some counterexamples: Gaussian elimination, …
But these algorithms tend to be "brittle"

Proof Techniques

How to rule out all low-degree polynomial algorithms?

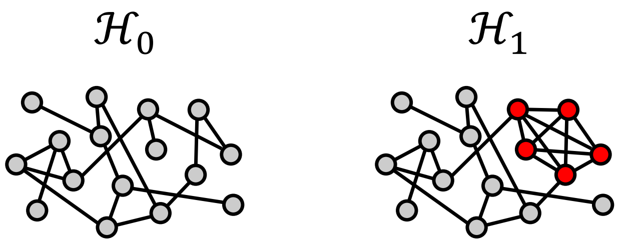
Depends on the type of problem…

- Hypothesis testing / Detection (is there a planted clique or not?) – next slide
  - Linear algebra [Hopkins-Steurer'17, HKPRSS17, …]
- Estimation / Recovery (there is a planted clique, find it) – part 2 of this talk
  - Linear algebra ++ [Schramm-**W**'20, …]
- Optimization (there is no planted clique, find a large clique)
  - Overlap gap property [Gamarnik-Jagannath-**W**'20, …]
- Refutation (there is no planted clique, prove there is no large clique)
  - Quiet planting, reduce to testing [Bandeira-Kunisky-**W**'19, …]

low-degree polynomial ↓ planted clique ↓

Example: LDP-Hardness of PC Detection

[Barak, Hopkins, Kelner, Kothari, Moitra, Potechin '16; Hopkins '18]



Goal: if $k \le n^{1/2-\epsilon}$ then for some $D = \omega(\log n)$, no degree-D

$$f : \{-1,1\}^{\binom{n}{2}} \to \mathbb{R}$$

(no edge ↑ -1, edge ↑ 1)

achieves "separation"

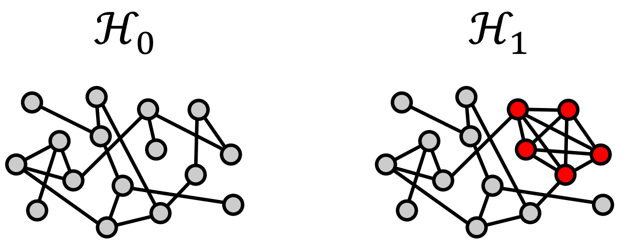
$$\sqrt{\max\{\mathrm{Var}_{\mathcal{H}_0}(f), \mathrm{Var}_{\mathcal{H}_1}(f)\}} = o\left(\mathrm{E}_{\mathcal{H}_1}[f] - \mathrm{E}_{\mathcal{H}_0}[f]\right).$$

Suffices to show:

$$\mathrm{Adv}_{\le D} := \sup_{f \deg D} \frac{\mathbb{E}_{\mathcal{H}_1}[f(A)]}{\sqrt{\mathbb{E}_{\mathcal{H}_0}[f(A)^2]}} = O(1)$$

Example: LDP-Hardness of PC Detection

[Barak, Hopkins, Kelner, Kothari, Moitra, Potechin ’16; Hopkins ‘18]



Suffices to show: $\mathrm{Adv}_{\le D} := \sup_{f \deg D} \frac{\mathbb{E}_{\mathcal{H}_1}[f(A)]}{\sqrt{\mathbb{E}_{\mathcal{H}_0}[f(A)^2]}} = O(1)$

$f : \{-1,1\}^{\binom{n}{2}} \to \mathbb{R}$ $\quad f(A) = \sum_{|S| \le D} \hat{f}_S A^S$ $\quad A^S := \prod_{(i,j)\in S} A_{ij}$

- Numerator linear in $\hat{f}$ $\quad \mathbb{E}_{\mathcal{H}_1}[f(A)] = \sum_S \hat{f}_S \underbrace{\mathbb{E}_{\mathcal{H}_1}[A^S]}_{c_S} =: \langle c, \hat{f} \rangle$
- Denominator quadratic in $\hat{f}$ $\quad \mathbb{E}_{\mathcal{H}_0}[f(A)^2] = \sum_{S,S'} \hat{f}_S \hat{f}_{S'} \underbrace{\mathbb{E}_{\mathcal{H}_0}[A^S A^{S'}]}_{\mathbb{1}_{S=S'}} = \|\hat{f}\|^2$

$$\mathrm{Adv}_{\le D} = \sup_{\hat{f}} \frac{\langle c, \hat{f} \rangle}{\|\hat{f}\|} = \|c\| = \sqrt{\sum_{|S| \le D} \left(\mathbb{E}_{\mathcal{H}_1}[A^S]\right)^2}$$

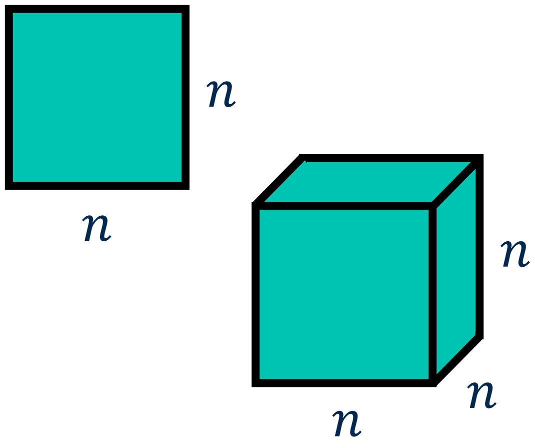
Part 2

Computational phase transitions in tensor decomposition

Tensors

Order-2 tensor: matrix $M = (M_{ij})$

Order-3 tensor $T = (T_{ijk})$



Rank-1 (symmetric) order-2 tensor $vv^\top$ $(vv^\top)_{ij} = v_i v_j$ $v \in \mathbb{R}^n$

Rank-1 (symmetric) order-3 tensor $v^{\otimes 3}$ $\left(v^{\otimes 3}\right)_{ijk} = v_i v_j v_k$

Tensor Decomposition

Basic primitive with applications in:

- Phylogenetic reconstruction [MR05]
- Topic modeling [AFHKL12]
- Community detection [AGHK13,HS17,AAA17,JLLX20]
- Learning Gaussian mixtures [HK13,GHK15,BCMV14,ABGRV14]
- Independent component analysis [GVX14]
- Dictionary learning [BKS15,MSS16]
- Multi-reference alignment [PWBRS19]
- …

Random Tensor Decomposition

Given a rank-r order-3 tensor

$$T = \sum_{i=1}^{r} a_i^{\otimes 3} \qquad a_i \in \mathbb{R}^n$$

the goal is to recover the components $a_1, \ldots, a_r$

Assume random components $a_i \sim \mathcal{N}(0, I_n)$

succeed with high probability

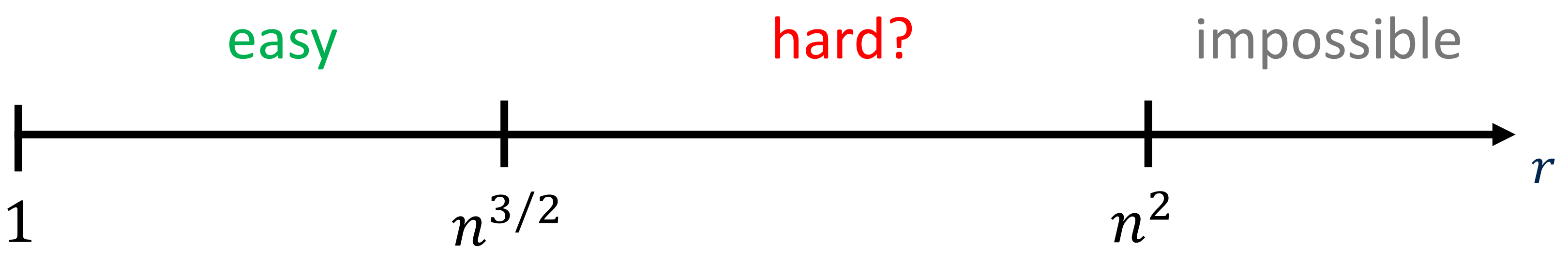
$$T = \sum_{i=1}^{r} a_i^{\otimes 3} \qquad a_i \sim \mathcal{N}(0, I_n)$$

Prior Work

Algorithmic results: SoS [GM15, Ma-Shi-Steurer'16], spectral [HSSS16,DOLST22], …

All known poly-time algorithms require $r \ll n^{3/2}$ $\ll$ hides polylog factor

Information-theoretically possible when $r \leq cn^2$ [BCO14] c = constant



Q: is this hardness inherent?

Statistical-Computational Gaps

Many statistical problems have “hard” regimes

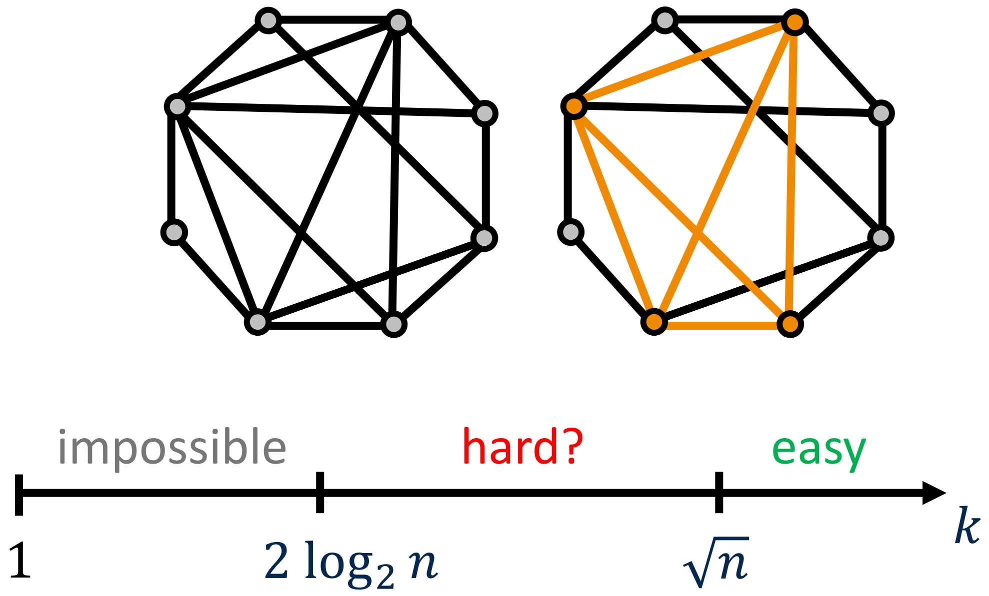
sparse PCA, compressed sensing, community detection, tensor PCA, …

No average-case complexity theory

Instead:

- Reductions from planted clique
- Lower bounds in restricted models
- Optimization landscape

Planted clique: G(n,1/2) + {k-clique}



Tensor Decomposition: Difficulties

Which lower bound framework?

- Reduction – out of reach?
- Statistical query (SQ) model – not applicable (no iid samples)
- Sum-of-squares (SoS) – hardness of refutation [BBKMW21]
- Optimization landscape – what function to optimize? [GZ19, BGJ20, CMZ22]
- Low-degree polynomials (LDP) – this talk

Tensor Decomposition: More Difficulties

Issue of symmetry

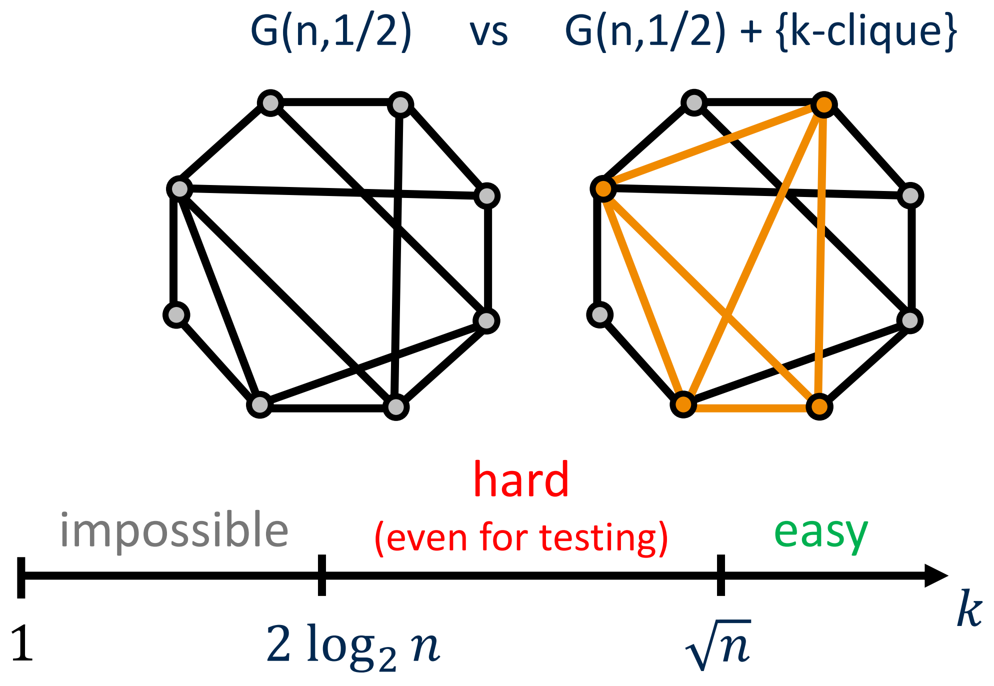
which component to recover?

$$T = \sum_{i=1}^{r} a_i^{\otimes 3}$$

Existing SQ/SoS/LDP lower bounds leverage hardness of testing vs iid "null"

a few exceptions [Schramm-**W**'22, Koehler-Mossel'21]

- Testing rank-r tensor vs iid tensor is only hard when $r \gg n^3$
- (Decomp hard when $r \gg n^{3/2}$)



Solving the Issue of Symmetry

Define a new model: "largest component recovery"

$$T = (1+\delta)a_1^{\otimes 3} + \sum_{i=2}^{r} a_i^{\otimes 3} \qquad a_i \in \{\pm 1\}^n \text{ unif. at random}$$

Goal: recover/estimate $a_{11} := (a_1)_1$

Hardness of the above problem implies hardness of decomposing

$$\sum_{i=1}^{r} \lambda_i a_i^{\otimes 3} \qquad \lambda_i \in [1, 1+\delta] \text{ arbitrary}, \qquad a_i \in \{\pm 1\}^n \text{ unif. at random}$$

Main Result: LDP Phase Transition

Class of algorithms: multivariate polynomials f in the entries of

$$T = (1+\delta)a_1^{\otimes 3} + \sum_{i=2}^{r} a_i^{\otimes 3} \qquad a_i \in \{\pm 1\}^n \text{ unif. at random}$$

Degree-D minimum mean squared error:

$$\mathrm{MMSE}_{\leq D} := \inf_{f \deg D} \mathbb{E}_a[(f(T) - a_{11})^2]$$

Theorem (W. '22) Fix any $\epsilon > 0,\ \delta > 0$

- *(Easy)* If $r \leq n^{3/2-\epsilon}$ then $\mathrm{MMSE}_{\leq O(\log n)} \to 0$ as $n \to \infty$
- *(Hard)* If $r \geq n^{3/2+\epsilon}$ then $\mathrm{MMSE}_{\leq n^{\Omega(1)}} \to 1$ as $n \to \infty$

Main Result: LDP Phase Transition

Class of algorithms: multivariate polynomials f in the entries of

$$T = (1+\delta)a_1^{\otimes 3} + \sum_{i=2}^{r} a_i^{\otimes 3} \qquad a_i \in \{\pm 1\}^n \text{ unif. at random}$$

Degree-D minimum mean squared error:

$$\mathrm{MMSE}_{\leq D} := \inf_{f \deg D} \mathbb{E}_a[(f(T) - a_{11})^2]$$

Theorem (W. '22) Fix any $\epsilon > 0,\ \delta > 0$

- *(Easy)* If $r \leq n^{3/2-\epsilon}$ then $\mathrm{MMSE}_{\leq O(\log n)} \to 0$ as $n \to \infty$
- *(Hard)* If $r \geq n^{3/2+\epsilon}$ then $\mathrm{MMSE}_{\leq n^{\Omega(1)}} \to 1$ as $n \to \infty$

Upper Bound: LDP Succeeds

Idea: “spectral methods from tensor networks”

[Hopkins-Schramm-Shi-Steurer’16, Moitra-**W**’19, Ding-d’Orsi-Liu-Steurer-Tiegel’22]

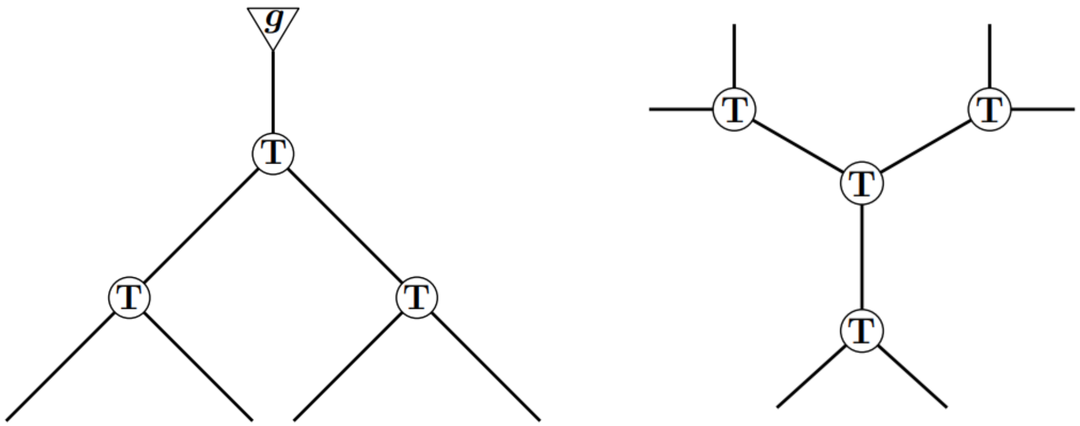
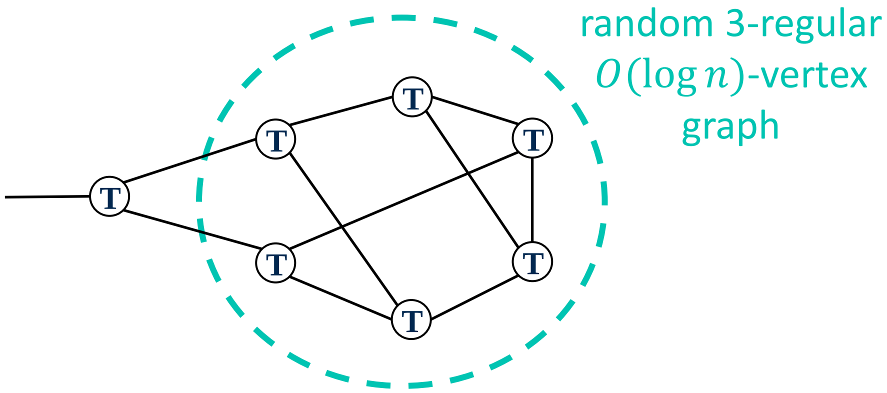


Image credit: [DOLST22]



This work

Degree-$O(\log n)$ polynomial implies quasipoly-time $n^{O(\log n)}$ algorithm

Main Result: LDP Phase Transition

Class of algorithms: multivariate polynomials f in the entries of

$$T = (1+\delta)a_1^{\otimes 3} + \sum_{i=2}^{r} a_i^{\otimes 3} \qquad a_i \in \{\pm 1\}^n \text{ unif. at random}$$

Degree-D minimum mean squared error:

$$\mathrm{MMSE}_{\leq D} := \inf_{f \deg D} \mathbb{E}_a[(f(T) - a_{11})^2]$$

Theorem (W. '22) Fix any $\epsilon > 0,\ \delta > 0$

- *(Easy)* If $r \leq n^{3/2-\epsilon}$ then $\mathrm{MMSE}_{\leq O(\log n)} \to 0$ as $n \to \infty$
- *(Hard)* If $r \geq n^{3/2+\epsilon}$ then $\mathrm{MMSE}_{\leq n^{\Omega(1)}} \to 1$ as $n \to \infty$

Lower Bound: Baby Example

Observe scalar $t = \sum_{i=1}^{r} a_i$ $a_i \in \{\pm 1\}$ unif. at random

Goal: estimate a_1

$$\sup_{\hat{f}} \frac{\langle c, \hat{f} \rangle}{\sqrt{\hat{f}^\top P \hat{f}}} = \sqrt{c^\top P^{-1} c}$$

//

Want to show $\mathrm{Corr}_{\leq D} \coloneqq \sup_{f \deg D} \frac{\mathbb{E}[f(t) a_1]}{\sqrt{\mathbb{E}[f(t)^2]}} = o(1)$ $f(t) = \sum_{d=0}^{D} \hat{f}_d t^d$

First attempt:

- Numerator linear in $\hat{f}$ $\mathbb{E}[f(t) a_1] = \sum_{d=0}^{D} \hat{f}_d \underbrace{\mathbb{E}[t^d a_1]}_{c_d} =: \langle c, \hat{f} \rangle$
- Denominator quadratic in $\hat{f}$ $\mathbb{E}[f(t)^2] = \sum_{d,d'=0}^{D} \hat{f}_d \hat{f}_{d'} \underbrace{\mathbb{E}[t^d t^{d'}]}_{P_{d,d'}} =: \hat{f}^\top P \hat{f}$

$$t = \sum_{i=1}^{r} a_i \qquad a_i \in \{\pm 1\}$$

want $\frac{\mathbb{E}[f(t)a_1]}{\sqrt{\mathbb{E}[f(t)^2]}} \leq \cdots$

Lower Bound: Baby Example

$$\sum_{d=0}^{D} \hat{f}_d t^d = f(t) = g(a) = \sum_{U \subseteq [r]} \hat{g}_U a^U \longleftarrow a^U := \prod_{i \in U} a_i$$

orthonormal: $\mathbb{E}\left[a^U a^{U'}\right] = \mathbb{1}_{U=U'}$

Claim: $\mathbb{E}[f(t)^2] = \mathbb{E}[g(a)^2] = \|\hat{g}\|^2$

Claim: $\hat{g} = M\hat{f}$ for some matrix M

Claim: suffices to construct an explicit left-inverse M^+ s.t. $M^+ M = I$

$$\sup_f \frac{\mathbb{E}[f(t)a_1]}{\sqrt{\mathbb{E}[f(t)^2]}} = \sup_{\hat{f}} \frac{\langle c, \hat{f} \rangle}{\|M\hat{f}\|} = \sup_{\hat{f}} \frac{c^\top M^+ M \hat{f}}{\|M\hat{f}\|} \leq \sup_{\hat{g}} \frac{c^\top M^+ \hat{g}}{\|\hat{g}\|} = \|c^\top M^+\|$$

$\uparrow$ $\|\hat{g}\|$

$$t = \sum_{i=1}^{r} a_i \qquad a_i \in \{\pm 1\}$$

Constructing the Left-Inverse

$$\sum_{d=0}^{D} \hat{f}_d t^d = f(t) = g(a) = \sum_{U \subseteq [r]} \hat{g}_U a^U$$

Recall: $\hat{g} = M\hat{f}$ want M^+ s.t. $M^+ M = I$

In other words: $M^+ \hat{g} = \hat{f}$ whenever $\hat{g} = M\hat{f}$

In other words: given (valid) $\hat{g}$, recover $\hat{f}$

$t = a_1 + a_2 + a_3 + a_4$

↓

Proof by example: $g(a) = a_1 a_2 a_3 + a_1 a_2 a_4 - 2a_1 a_3 - 2a_3 a_4 + \cdots$ $\qquad f(t) = ??$

$r = 4, D = 3$

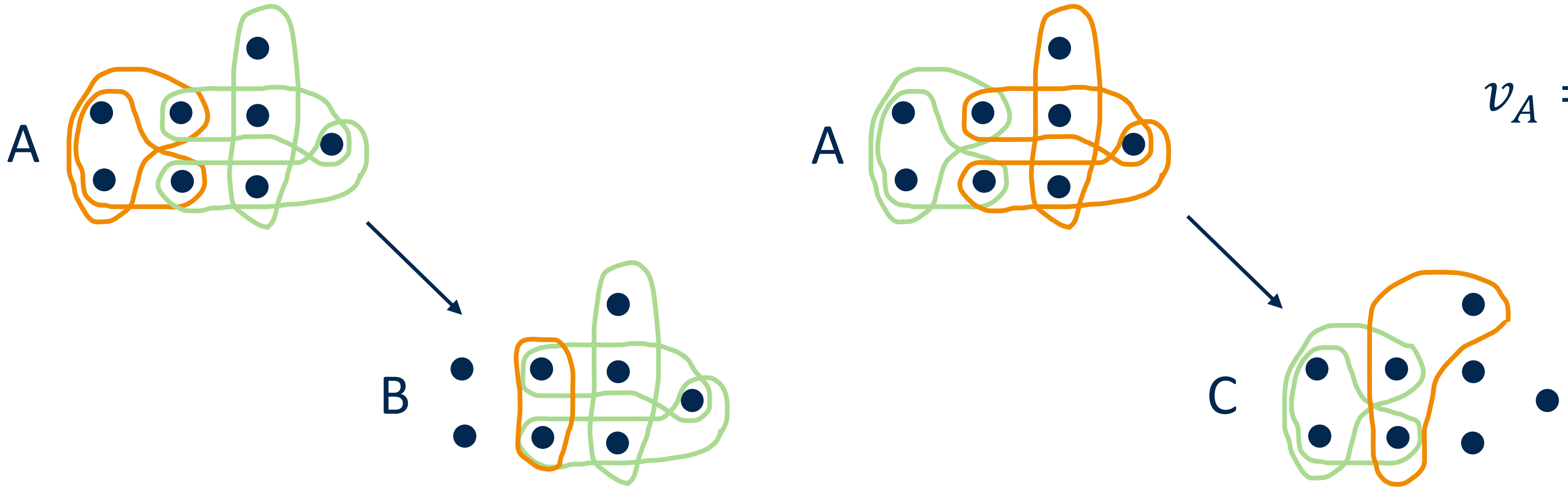
$$\frac{1}{6} t^3 = a_1 a_2 a_3 + a_1 a_2 a_4 + \cdots + \frac{5}{3}(a_1 + a_2 + a_3 + a_4)$$

Wrapping Up the Lower Bound

Conclusion: $\mathrm{Corr}_{\leq D} \coloneqq \sup_{f \deg D} \frac{\mathbb{E}[f(t)a_1]}{\sqrt{\mathbb{E}[f(t)^2]}} \leq \|c^\top M^+\| =: \|v\|$

For the true model, v is indexed by hypergraphs and defined recursively

reminiscent of cumulants in [Schramm-**W**'22]



$$v_A = \alpha_{AB} v_B + \alpha_{AC} v_C + \cdots$$

Thanks!

Comments

First concrete lower bound for random tensor decomposition

low-degree polynomial threshold matches best known algorithms

Results extend to tensors of order $k \geq 3$, threshold is $r \sim n^{k/2}$

Future directions: Gaussian components, structured tensors

Open: is “generic” tensor decomposition strictly harder than random ($k = 3$)?