EFFICIENT BAYESIAN INFERENCE, PLANTED PROBLEMS, AND SUM OF SQUARES ALGORITHMS

Sam Hopkins (Cornell)

based on the works

A nearly-tight sum-of-squares lower bound for the planted clique problem.

[Barak-**H.**-Kelner-Kothari-Moitra-Potechin, FOCS 2016]

The power of SoS for detecting hidden structures.

[**H.**-Kothari-Potechin-Raghavendra-Schramm-Steurer] (available soon)

Efficient Bayesian estimation from few samples: community detection and related problems.

[**H.**-Steurer] (available soon)

PLANTED PROBLEMS

nature samples $x \sim \mathcal{U}$, then $y \sim p(y|x)$
see y, try to recover x

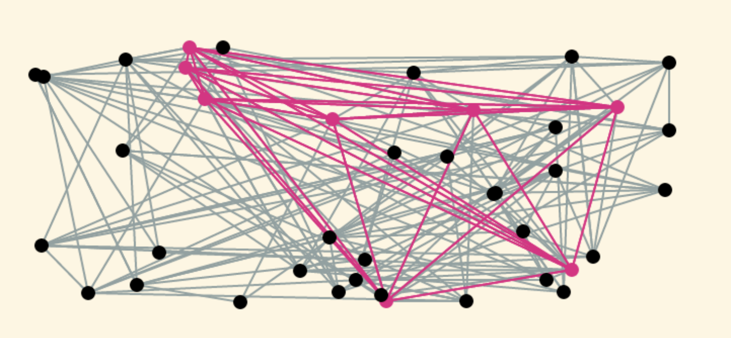
Usually, $\mathcal{U} = \{\pm 1\}^n, \mathbf{R}^n, \ldots$
y is a graph or matrix or CNF formula or …

This talk:
simple *low degree tests* criterion *determines algorithmic difficulty* of given planted problem
Developing picture: (partly/largely conjectural)
criterion satisfied → generic meta-algorithm solves efficiently (“Bayesian SoS”)
criterion not satisfied → SoS algorithms fail

CSP(P): P is a boolean predicate, $x \in \{\pm 1\}^n$ and y is an m-clause random instance of P satisfied by x

$$y = (x_1 \vee x_{10} \vee \neg x_{27}) \wedge (\neg x_{19} \vee x_4 \vee \neg x_{12})$$

planted/hidden k-clique: $x \subset [n]$ has size k and $y \sim \mathbb{G}(n, \frac{1}{2})$ conditioned on x a clique in y.



sparse PCA: $x \in \mathbf{R}^n$ is a random k-sparse unit vector and $y = y_1, \dots, y_m \sim N(0, I + xx^\top)$.

WHY STUDY PLANTED PROBLEMS?

natural average-case versions of combinatorial optimization problems

(toy) models for fundamental statistics problems

source of interesting instances

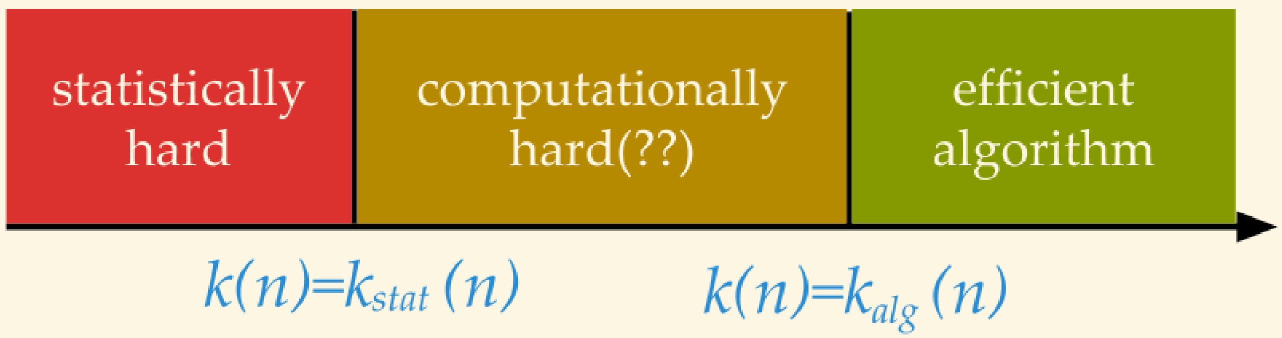
where the hard instances are?

where the easy instances are?

TYPICAL ALGORITHMIC LANDSCAPE

Planted problems come with a parameter to vary hardness

CSP: $m = m(n)$ for m-clause n-variate CSP
planted clique: $k = k(n)$ for k-clique in n-node graph
sparse PCA: $m = m(k, n)$ samples from Gaussian with k-sparse spike



planted clique: $k_{stat}(n) = 2\log n$ but $k_{alg}(n) = \Theta(\sqrt{n})$
HOWEVER k_{alg} *is always conjectural*

RIGOROUS EVIDENCE FOR COMPUTATIONAL HARDNESS OF PLANTED PROBLEMS?

$3SAT \leqslant_p$ planted clique?

unlikely: gadget reductions break specific distribution on instances

Instead, prove *unconditional results* for *restricted models*.

Examples:

1. Efficient Markov-Chain Monte-Carlo algorithms, polynomial-size Lovasz-Schrijver+ relaxations cannot find $o(n^{1/2})$-size cliques n-node in random graphs [Jerrum 92, Feige-Krauthgamer 03]

2. Basic SDP and degree-4 SoS relaxations of sparse PCA cannot tolerate fewer than $m = (\text{sparsity})^2$ samples [Krauthgamer-Nadler-Vilenchik 15, Ma-Wigderson 15]

SOS IS THE FRONTIER

rule out stronger algorithms → better evidence for computationally-hard region

for many problems, $n^{O(1)}$-time SoS *succeeds against harder parameters* than other known algorithms.

e.g. **tensor pca**, dictionary learning, random tensor decomposition, sparse vector in random subspace

[Barak-Kelner-Steurer 15, Ge-Ma 15, Ma-Shi-Steurer 16, **H.**-Shi-Steurer 15]

understanding when SoS succeeds/fails is critical to understanding k_{alg} for planted problems:

$$k_{SoS} \geqslant k_{alg}$$

"SoS is Optimal" conjecture: $k_{alg} \approx k_{SoS}$

TYPICAL QUESTION

For constant d and $G \sim \mathbb{G}(n, 1/2)$, is $\mathrm{SoS}_d(G) \geqslant n^{1/2-o(1)}$?
$\mathrm{SoS}_d(G)$ is the degree-d SoS relaxation of

$$\max \sum_{i=1}^{n} x_i \quad \text{s.t. } x_i^2 = x_i \text{ and } x_i x_j = 0 \text{ if } i \nsim j$$

Canonical SoS relaxation for *Max-Clique*, natural SoS algorithm for planted clique.

Resolved in line of work

[Meka-Potechin-Wigderson 15]

[Deshpande-Montanari 15]

[**H.**-Kothari-Potechin-Raghavendra-Schramm 16]

[Barak-**H.**-Kelner-Kothari-Moitra-Potechin 16]

REST OF TALK

1. Study *low-degree tests/estimators*: **simple** and easily-analyzed algorithms tailored to planted problems.
   (e.g. average degree)
2. Relate *best low-degree estimator* to SoS.

Benefit A (if you like planted problems): enough to analyze low-degree estimators to make excellent guess for k_{alg}

Benefit B (if you like SoS/meta-algorithms): strong indication that SoS performance $\approx$ performance of low-degree estimators, *even though SoS not tailored to the setting*.

WHAT IS A LOW-DEGREE TEST?

Two hypotheses:

$H_0 : G \sim \mathbb{G}(n, 1/2)$

$H_1 : G \sim \mathbb{G}(n, 1/2) + k$-clique

A good low degree test is $\alpha(G)$: graphs $\to \mathbb{R}$ with

1. $\deg(\alpha) \leqslant D$
2. $\mathbb{E}_{G\sim\mathbb{G}(n,1/2)} \alpha(G) = 0$
3. $\mathbb{E}_{G\sim\mathbb{G}(n,1/2)} \alpha(G)^2 = 1$

$$\mathbb{E}_{G\sim\text{planted}}\ \alpha(G) \geqslant \delta = \Omega_n(1).$$

Example: $\alpha(G) =$ # of triangles in G.

"Theorem" (Cauchy-Schwarz): optimal δ is

$$\delta_{\text{best}} = \left\| \left(\frac{\mathbb{P}_{\text{planted}}(G)}{\mathbb{P}_{\mathbb{G}(n,1/2)}(G)} \right)^{\leqslant D} \right\|$$

(which can be computed with simple linear algebra/Fourier analysis).

graph problems: degree-D tests = D-edge subgraph statistics
CSPs: degree-D tests = bounded-width resolution refutations
(or D-hyperedge subgraphs of clause hypergraph)

For constant (or logarithmic) D,
planted clique: $\delta_{\text{best}} \geqslant \Omega_n(1)$ iff $k \geqslant \sqrt{n}$
sparse PCA: $\delta_{\text{best}} \geqslant \Omega_n(1)$ iff $m \geqslant (\text{sparsity})^2$

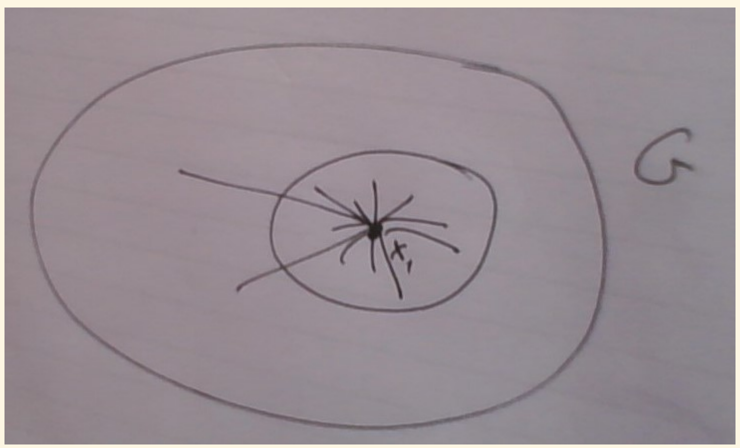
LOW DEGREE ESTIMATORS

$H_0 : \mathbb{G}(n, 1/2)$
$H_1 : \mathbb{G}(n, 1/2) + k$-clique
(applies also to sparse pca, tensor pca, stochastic blockmodels, etc.)

A good *low degree estimator* of $x_1(G)$, (normalized indicator that vertex 1 is in the clique), is

1. $\alpha(G)$: graphs $\to \mathbb{R}$
2. $\deg(\alpha) \leqslant D$
3. $\mathbb{E}_{\mathbb{G}(n,1/2)} \alpha(G)^2 = 1$

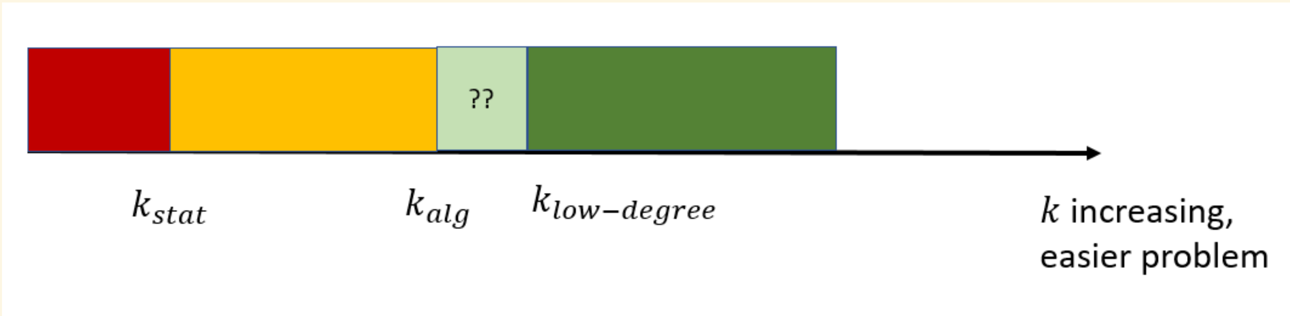
and $\mathbb{E}_{x,G\sim\text{planted}} \ \alpha(G) \cdot x_1 \geqslant \delta$ (e.g. $\alpha(G) = \deg(\text{vertex } 1)$)



[H.-Steurer]: low-degree estimators + SoS tensor decomposition unify *many planted problem algorithms* with *generic analysis*.
for numerous problems $k_{alg} = k_{low-deg}$

LOW-DEGREE ESTIMATORS AND SOS

Taking stock: $k_{stat} < k_{alg} \leqslant k_{low-degree}$



Next: $k_{low-degree} \leqslant k_{sos}$ (“pseudocalibration”)

For constant d and $G \sim \mathbb{G}(n, 1/2)$, is $\mathrm{SoS}_d(G) \geqslant n^{1/2 - o(1)}$?
$\mathrm{SoS}_d(G)$ is the degree-d SoS relaxation of

$$\max \sum_{i=1}^{n} x_i \quad \text{s.t. } x_i^2 = x_i \text{ and } x_i x_j = 0 \text{ if } i \nsim j$$

Initial difficulty: how to define, for typical $G \sim \mathbb{G}(n, 1/2)$, a pseudodistribution on "large cliques" in G?

What should $\tilde{\mathbb{E}} x_i x_j x_k x_\ell$ *be?*

Idea 1: If $G \sim \mathbb{G}(n, 1/2) + k$-clique, then degree-$d$ SoS is trying to compute $\mathbb{E}x^{\otimes d}|G$.

Idea 2 (wild guess!!): SoS computes a *low-degree estimate* of $\mathbb{E}x^{\otimes d}|G$.

Idea 3: $\tilde{\mathbb{E}}x_i x_j x_k x_\ell = (\text{scaling}) \cdot \alpha_{ijk\ell}(G)$ for best low-degree estimator α.

Fourier analysis: Every low-degree $f(G,x)$ is fooled by this $\tilde{E}$:

$$\mathbb{E}_{x,G\sim\text{planted}}\ f(G,x) = \mathbb{E}_{\mathbb{G}(n,1/2)}\tilde{E}f(G,x)$$

In particular, $\tilde{E}\sum x_i = k$.

no low-degree estimator → $Var_{\mathbb{G}(n,1/2)}(\tilde{E}x_i x_j x_k x_\ell)$ "small"

Theorem [many papers]**:** For planted clique, sparse PCA, random CSPs, tensor PCA, $\tilde{E}$ is a pseudodistribution whp.

CLOSING REMARKS AND OPEN PROBLEMS

$k_{\text{stat}} < k_{\text{alg}} \leqslant k_{\text{low-degree}}$

For many problems, $k_{\text{low-degree}} \leqslant k_{\text{SoS}}$.

Conjecture: $k_{\text{alg}} = k_{\text{low-degree}}$: efficient algorithms *tailored to each individual planted problem* ONLY compute low-degree statistics

Conjecture: $k_{\text{low-degree}} = k_{\text{SoS}}$: SoS does as well as best algorithm tailored to particular problem, but SoS is "generic"

More open problems:

1. SoS lower bound for densest-k-subgraph or other sparse problem
2. Meta-theorem about SoS versus low-degree algorithms?
3. Separate SoS and low-degree algorithms for some planted problem?

THANK YOU!