Finding a Large Submatrix of a Random Matrix, and the Overlap Gap Property

David Gamarnik

MIT

The Classification Program of Counting Complexity

Joint work with Quan Li (MIT)

April 1, 2016

- Many optimization problems over random instances (random K-SAT, coloring of a random graph, maximum independent set of a random graph) exhibit an apparent gap between algorithmic and existential results.

- Many optimization problems over random instances (random K-SAT, coloring of a random graph, maximum independent set of a random graph) exhibit an apparent gap between algorithmic and existential results.
- What is the source of the apparent hardness?

- Many optimization problems over random instances (random K-SAT, coloring of a random graph, maximum independent set of a random graph) exhibit an apparent gap between algorithmic and existential results.
- What is the source of the apparent hardness?
- *Overlap Gap Property* originating from the theory of Spin Glasses.

- Many optimization problems over random instances (random K-SAT, coloring of a random graph, maximum independent set of a random graph) exhibit an apparent gap between algorithmic and existential results.

- What is the source of the apparent hardness?

- *Overlap Gap Property* originating from the theory of Spin Glasses.

- This talk: illustration of the OGP using the maximum submatrix problem.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $\sim 2 \log_{\frac{1}{p}} n$.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $\sim 2 \log_{\frac{1}{p}} n$.
- A trivial greedy algorithm finds a clique of size $\sim \log_{\frac{1}{p}} n$.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $\sim 2\log_{\frac{1}{p}} n$.
- A trivial greedy algorithm finds a clique of size $\sim \log_{\frac{1}{p}} n$.
- Karp [1976] Find a better algorithm.

(Arguably) Most Embarrassing Algorithmic Problem in Random Graphs

- Consider $\mathbb{G}(n, p)$.
- The largest clique (fully connected subgraph) is $\sim 2 \log_{\frac{1}{p}} n$.
- A trivial greedy algorithm finds a clique of size $\sim \log_{\frac{1}{p}} n$.
- Karp [1976] Find a better algorithm.
- Still open. This is embarrassing...

Sparse Random Graphs: Similar Story

Sparse Random Graphs: Similar Story

- Consider $\mathbb{G}(n, d/n)$. d is constant.

Sparse Random Graphs: Similar Story

- Consider $\mathbb{G}(n, d/n)$. d is constant.
- The largest independent set (a subset of nodes with no edges) is

$$\sim \frac{2 \log d}{d} n,$$

Frieze, Luczak [1992]

Sparse Random Graphs: Similar Story

- Consider $\mathbb{G}(n, d/n)$. d is constant.
- The largest independent set (a subset of nodes with no edges) is

$$\sim \frac{2 \log d}{d} n,$$

Frieze, Luczak [1992]

- A trivial greedy algorithm finds an independent set of size

$$\sim \frac{\log d}{d} n.$$

Sparse Random Graphs: Similar Story

- Consider $\mathbb{G}(n, d/n)$. d is constant.
- The largest independent set (a subset of nodes with no edges) is

$$\sim \frac{2 \log d}{d} n,$$

  Frieze, Luczak [1992]

- A trivial greedy algorithm finds an independent set of size

$$\sim \frac{\log d}{d} n.$$

- Better algorithm?

Sparse Random Graphs: Similar Story

- Consider $\mathbb{G}(n, d/n)$. d is constant.
- The largest independent set (a subset of nodes with no edges) is

$$\sim \frac{2\log d}{d} n,$$

Frieze, Luczak [1992]

- A trivial greedy algorithm finds an independent set of size

$$\sim \frac{\log d}{d} n.$$

- Better algorithm?
- Similar story for many other combinatorial optimization problems.

Given $n \times n$ matrix C_n with standard normal i.i.d. entries

$$C_n = \begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & \ddots & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix},$$

and given k, find a $k \times k$ submatrix

$$\begin{bmatrix} C_{11} & \dots & C_{1n} \\ \vdots & \begin{bmatrix} * & \dots & * \\ \vdots & C^*_{n,k} & \vdots \\ * & \dots & * \end{bmatrix} & \vdots \\ C_{n1} & \dots & C_{nn} \end{bmatrix},$$

with the largest average entry

$$\mathrm{Ave}(C^*_{n,k}) = \frac{1}{k^2} \sum_{1 \le c,r \le k} C_{i_c,j_r}$$

Paper Motivating This Works

Paper Motivating This Works

[Bhamidi, Dey & Nobel 2013]

Paper Motivating This Works

[Bhamidi, Dey & Nobel 2013]

- Global optimum

$$\mathrm{Ave}(C^*_{n,k}) \approx \sqrt{\frac{4\log n}{k}}.$$

Paper Motivating This Works

[Bhamidi, Dey & Nobel 2013]

- Global optimum

$$\mathrm{Ave}(C^*_{n,k}) \approx \sqrt{\frac{4\log n}{k}}.$$

- Intuition:

Paper Motivating This Works

[Bhamidi, Dey & Nobel 2013]

- Global optimum

$$\mathrm{Ave}(C^*_{n,k}) \approx \sqrt{\frac{4 \log n}{k}}.$$

- Intuition:
  - If $Z_1, \ldots, Z_N$ are $N(0, \sigma^2)$, then $\max Z_i \approx \sigma\sqrt{2 \log N}$.

Paper Motivating This Works

[Bhamidi, Dey & Nobel 2013]

- Global optimum

$$\mathrm{Ave}(C^*_{n,k}) \approx \sqrt{\frac{4\log n}{k}}.$$

- Intuition:
  - If $Z_1, \ldots, Z_N$ are $N(0, \sigma^2)$, then $\max Z_i \approx \sigma\sqrt{2\log N}$.
  - Maximum of $\binom{n}{k}^2$ $N(0, \frac{1}{k^2})$ Gaussians is then

$$\frac{1}{k}\sqrt{2\log\binom{n}{k}^2} \approx \frac{1}{k}\sqrt{4k\log n}.$$

Paper Motivating This Works

[Bhamidi, Dey & Nobel 2013]

- Global optimum

$$\mathrm{Ave}(C^*_{n,k}) \approx \sqrt{\frac{4 \log n}{k}}.$$

- Intuition:
  - If $Z_1, \ldots, Z_N$ are $N(0, \sigma^2)$, then $\max Z_i \approx \sigma\sqrt{2 \log N}$.
  - Maximum of $\binom{n}{k}^2$ $N(0, \frac{1}{k^2})$ Gaussians is then

$$\frac{1}{k}\sqrt{2 \log \binom{n}{k}^2} \approx \frac{1}{k}\sqrt{4k \log n}.$$

- What about algorithms?

Motivation and Prior Work

Motivation and Prior Work

- Genetics, bioinformatics and social networks. Madeira [2004], Fortunato [2010], Shabalin [2009].

Motivation and Prior Work

- Genetics, bioinformatics and social networks. Madeira [2004], Fortunato [2010], Shabalin [2009].
- The problem of finding the optimal $k \times k$ submatrix amongst $\binom{n}{k}^2$ choices is computationally challenging for large k.

Motivation and Prior Work

- Genetics, bioinformatics and social networks. Madeira [2004], Fortunato [2010], Shabalin [2009].
- The problem of finding the optimal $k \times k$ submatrix amongst $\binom{n}{k}^2$ choices is computationally challenging for large k.
- A natural heuristics: **ISP** (Iterative Search Procedure) Shabalin [2009]. It iteratively updates rows and columns until no further improvement can be obtained.

ISP Algorithm

ISP Algorithm

ISP Algorithm

Input: An $n \times n$ matrix C and a fixed integer $k \geq 1$.

ISP Algorithm

Input: An $n \times n$ matrix C and a fixed integer $k \geq 1$.

Initialize: Select k columns J uniformly at random.

ISP Algorithm

Input: An $n \times n$ matrix C and a fixed integer $k \geq 1$.

Initialize: Select k columns J uniformly at random.

Loop: Iterate until convergence of I and J:
  Let $I := k$ rows with the largest entry sums over the columns in J.
  Let $J := k$ columns with the largest entry sums over the rows in I.

ISP Algorithm

Input: An $n \times n$ matrix C and a fixed integer $k \geq 1$.

Initialize: Select k columns J uniformly at random.

Loop: Iterate until convergence of I and J:
- Let $I := k$ rows with the largest entry sums over the columns in J.
- Let $J := k$ columns with the largest entry sums over the rows in I.

Output: Submatrix associated with final I and J.

- Observe that **ISP** outputs a matrix which is *locally* (row and column) optimal.
[Bhamidi, Dey & Nobel]: Most locally optimal matrices have value $1/\sqrt{2}$ smaller than the global optimum:

$$(1 + o(1))\sqrt{\frac{2 \log n}{k}}.$$

- Observe that **ISP** outputs a matrix which is *locally* (row and column) optimal.
  [Bhamidi, Dey & Nobel]: Most locally optimal matrices have value $1/\sqrt{2}$ smaller than the global optimum:

$$(1+o(1))\sqrt{\frac{2\log n}{k}}.$$

- Intuition. The best submatrix for a fixed set of k rows has average $\sim \sqrt{2\log n/k}$. Further iterations do not improve the average significantly.

- Open questions:

- Open questions:
- What is the value $\mathrm{Ave}(C_{n,k}^{\mathbf{ISP}})$ produced by **ISP**?

- Open questions:
- What is the value $\mathrm{Ave}(C_{n,k}^{\mathbf{ISP}})$ produced by **ISP**?
- Are there better algorithms?

- Open questions:
- What is the value $\text{Ave}(C_{n,k}^{\textbf{ISP}})$ produced by **ISP**?
- Are there better algorithms?
- What is the reason for apparent computational complexity?

A Simpler Algorithm

A Simpler Algorithm

- Fix $\theta > 0$ and let $A_{i,j} = \mathbf{1}\left(C_{i,j} > \theta\right)$.

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix},$$

A Simpler Algorithm

- Fix $\theta > 0$ and let $A_{i,j} = \mathbf{1}\left(C_{i,j} > \theta\right)$.

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix},$$

- A is a bi-partite Erdös-Rényi graph $\mathbb{G}(n, n, p), p = \mathbb{P}(Z > \theta)$.

A Simpler Algorithm

- Fix $\theta > 0$ and let $A_{i,j} = \mathbf{1}\left(C_{i,j} > \theta\right)$.

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix},$$

- A is a bi-partite Erdös-Rényi graph $\mathbb{G}(n, n, p), p = \mathbb{P}(Z > \theta)$.
- **Fact:** W.h.p. a simple greedy algorithm produces a $m \times m$ clique $A_{n,m}^{\textbf{Greedy},\theta}$ with $m = \log n / \log(1/p)$.

A Simpler Algorithm

- Fix $\theta > 0$ and let $A_{i,j} = \mathbf{1}\left(C_{i,j} > \theta\right)$.

$$A = \begin{bmatrix} 0 & 1 & 0 & \dots & 1 \\ 1 & 1 & 0 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 1 & \dots & 0 \end{bmatrix},$$

- A is a bi-partite Erdös-Rényi graph $\mathbb{G}(n, n, p), p = \mathbb{P}(Z > \theta)$.
- **Fact:** W.h.p. a simple greedy algorithm produces a $m \times m$ clique $A_{n,m}^{\textbf{Greedy},\theta}$ with $m = \log n / \log(1/p)$.
- **Observation:** The corresponding matrix $C_{n,m}^{\textbf{Greedy},\theta}$ has *minimum* entry value θ.

Main Results

Main Results

Theorem 1

*The number of iterations T_n of **ISP** is $O(1)$ and w.h.p.*

$$Ave(C_{n,k}^{\boldsymbol{ISP}}) = (1+o(1))\sqrt{\frac{2\log n}{k}}, \qquad \textit{factor } \sqrt{2} \textit{ smaller than } Ave(C_{n,k}^*)$$

Main Results

Theorem 1

*The number of iterations T_n of **ISP** is $O(1)$ and w.h.p.*

$$Ave(C_{n,k}^{\textbf{ISP}}) = (1 + o(1))\sqrt{\frac{2 \log n}{k}}, \qquad \text{factor } \sqrt{2} \text{ smaller than } Ave(C_{n,k}^{*})$$

Theorem 2

Setting $\theta = \sqrt{2 \log n/k}$, leads to $k \times k$ clique. Thus

$$C_{n,k}^{\textbf{Greedy},\theta} = (1 + o(1)) C_{n,k}^{\textbf{ISP},\theta}.$$

Main Results (continued)

Main Results (continued)

We propose a new algorithm *Sequential Greedy* (IG).

Main Results (continued)

We propose a new algorithm *Sequential Greedy* (IG).

Theorem 3

$$Ave(C_{n,k}^{\boldsymbol{SG}}) = (1 + o(1))\frac{4}{3}\sqrt{\frac{2\log n}{k}} = (1 + o(1))\frac{4}{3}Ave(C_{n,k}^{\boldsymbol{ISP}}).$$

Sequential Greedy Algorithm

Sequential Greedy Algorithm

- In even step $2t$ the algorithm produces greedily a $(t+1)\times t$ matrix $C^{\mathbf{SG}}_{n,t}$.

$$\begin{bmatrix} C_{11} & & \cdots & & C_{1n} \\ C_{21} & & t & & C_{2n} \\ \vdots & t+1 & \begin{bmatrix} * & \cdots & * \\ \vdots & C^{\mathbf{SG}}_{n,t} & \vdots \\ * & \cdots & * \end{bmatrix} & & \vdots \\ C_{n1} & & \cdots & & C_{nn} \end{bmatrix},$$

Sequential Greedy Algorithm

- In even step $2t$ the algorithm produces greedily a $(t+1) \times t$ matrix $C_{n,t}^{\mathbf{SG}}$.

$$\begin{bmatrix} C_{11} & & \dots & & C_{1n} \\ C_{21} & & t & & C_{2n} \\ \vdots & t+1 & \begin{bmatrix} * & \dots & * \\ \vdots & C_{n,t}^{\mathbf{SG}} & \vdots \\ * & \dots & * \end{bmatrix} & & \vdots \\ C_{n1} & & \dots & & C_{nn} \end{bmatrix},$$

- In odd step $2t+1$ it produces greedily a $(t+1) \times (t+1)$ matrix $C_{n,t}^{\mathbf{SG}}$.

Proof Sketch

Proof Sketch

- In step $2t$ ($2t+1$) the added column (row) has entry sum $\approx \sqrt{2t\log n}$.

Proof Sketch

- In step $2t$ ($2t+1$) the added column (row) has entry sum $\approx \sqrt{2t\log n}$.
- Thus the total value is

$$\sum_{1\le t\le k} 2\sqrt{2t\log n} \approx 2\sqrt{2\log n}\int_1^k t^{\frac{1}{2}}\,dt$$

$$= 2\sqrt{2\log n}\frac{2}{3}k^{\frac{3}{2}}$$

$$= \frac{4}{3}k\sqrt{2k\log n}$$

Proof Sketch

- In step $2t$ $(2t + 1)$ the added column (row) has entry sum $\approx \sqrt{2t \log n}$.
- Thus the total value is

$$\sum_{1 \le t \le k} 2\sqrt{2t \log n} \approx 2\sqrt{2 \log n} \int_1^k t^{\frac{1}{2}} dt$$
$$= 2\sqrt{2 \log n}\frac{2}{3}k^{\frac{3}{2}}$$
$$= \frac{4}{3}k\sqrt{2k \log n}$$

- Thus

$$\text{Ave}(C_{n,k}^{\mathbf{SG}}) \approx \frac{4}{3}\sqrt{\frac{2 \log n}{k}}$$

Overlap Gap Property (OGP) and Phase Transition

Overlap Gap Property (OGP) and Phase Transition

- Global optimum matrix:

$$\sqrt{2}\sqrt{\frac{2\log n}{k}}.$$

Overlap Gap Property (OGP) and Phase Transition

- Global optimum matrix:

$$\sqrt{2}\sqrt{\frac{2\log n}{k}}.$$

- Best one found algorithmically

$$\frac{4}{3}\sqrt{\frac{2\log n}{k}}$$

Overlap Gap Property (OGP) and Phase Transition

- Global optimum matrix:

$$\sqrt{2}\sqrt{\frac{2\log n}{k}}.$$

- Best one found algorithmically

$$\frac{4}{3}\sqrt{\frac{2\log n}{k}}$$

- What is happening in $[\frac{4}{3}, \sqrt{2}]$?

Overlap Gap Property (OGP) and Phase Transition

Overlap Gap Property (OGP) and Phase Transition

Fix $\alpha \in [\frac{4}{3}, \sqrt{2}]$ and two submatrices C_1, C_2 with

$$\text{Ave}(C_1) \approx \text{Ave}(C_2) \approx \alpha\sqrt{\frac{2\log n}{k}}.$$

Overlap Gap Property (OGP) and Phase Transition

Fix $\alpha \in [\frac{4}{3}, \sqrt{2}]$ and two submatrices C_1, C_2 with

$$\mathrm{Ave}(C_1) \approx \mathrm{Ave}(C_2) \approx \alpha\sqrt{\frac{2\log n}{k}}.$$

Theorem 4

The expected number of such pairs C_1, C_2 with $y_1 k$ common rows and $y_2 k$ common columns is

$$\exp\left(f\left(\alpha, y_1, y_2\right) k \log n\right),$$

where

$$f(\alpha, y_1, y_2) = 4 - y_1 - y_2 - \frac{2}{1 + y_1 y_2}\alpha^2.$$

Overlap Gap Property (OGP) and Phase Transition

Fix $\alpha \in [\frac{4}{3}, \sqrt{2}]$ and two submatrices C_1, C_2 with

$$\mathrm{Ave}(C_1) \approx \mathrm{Ave}(C_2) \approx \alpha\sqrt{\frac{2\log n}{k}}.$$

Theorem 4

The expected number of such pairs C_1, C_2 with $y_1 k$ common rows and $y_2 k$ common columns is

$$\exp\left(f\left(\alpha, y_1, y_2\right) k \log n\right),$$

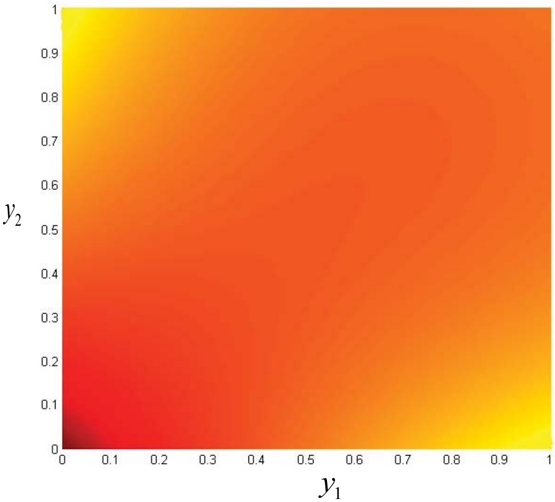
where

$$f(\alpha, y_1, y_2) = 4 - y_1 - y_2 - \frac{2}{1 + y_1 y_2}\alpha^2.$$

$f(\alpha, y_1, y_2) < 0$ implies no such pairs.

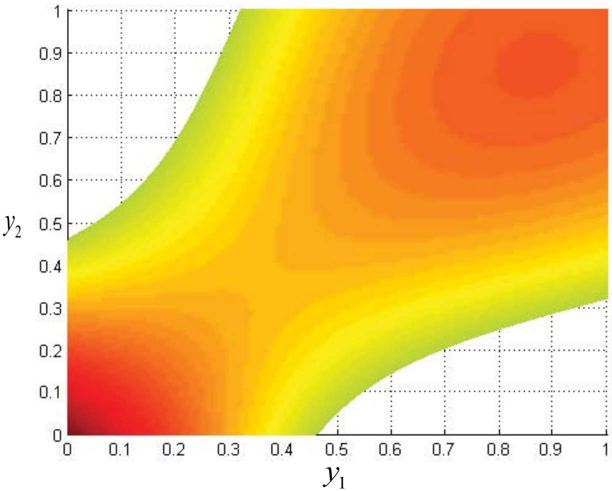
$\alpha < \alpha_* = \sqrt{3}/\sqrt{2} = 1.2247$.

$f(\alpha, y_1, y_2) > 0$ everywhere

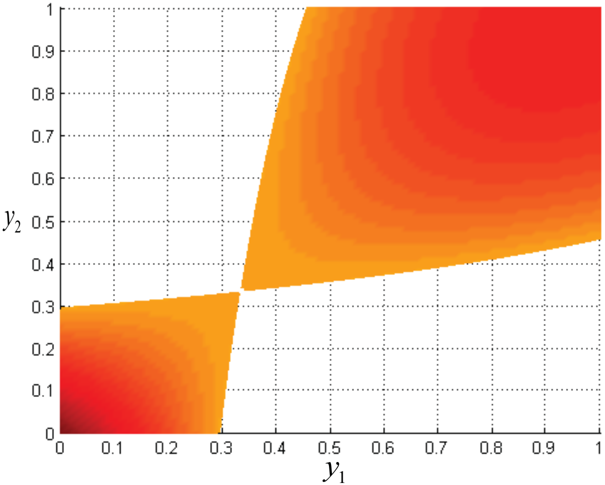


$\alpha_* < \alpha < \alpha^* = \frac{5}{3}\sqrt{\frac{2}{3}} = 1.3608$. **Includes** $4/3$

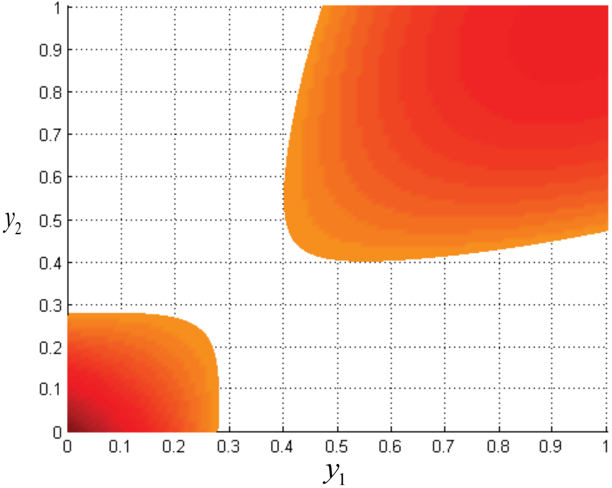
Color $f(\alpha, y_1, y_2) > 0$, white $f(\alpha, y_1, y_2) < 0$



At $\alpha^* = \frac{5}{3}\sqrt{\frac{2}{3}} = 1.3608$. Onset of the Overlap Gap Property



$\alpha \in [\alpha^*, \sqrt{2}]$. Overlap Gap Property



Limits for Local Algorithms

- The Overlap Gap Property occurs in sparse random graphs $\mathbb{G}(n, d/n)$ and is a *provable* obstacle for so-called *local* algorithms:

Theorem 5

[G & Sudan 2014, Rahman & Virag 2014] The largest independent set problem exhibits the Overlap Gap Property. As a result no local algorithm (appropriately defined) can improve upon the greedy algorithm.

- The Overlap Gap Property occurs in sparse random graphs $\mathbb{G}(n, d/n)$ and is a *provable* obstacle for so-called *local* algorithms:

Theorem 5

[G & Sudan 2014, Rahman & Virag 2014] The largest independent set problem exhibits the Overlap Gap Property. As a result no local algorithm (appropriately defined) can improve upon the greedy algorithm.

- Recall: in a sparse random graph $\mathbb{G}(n, \frac{d}{n})$

$$I^* \sim \frac{2 \log d}{d} n \qquad \text{vs} \qquad I^{\text{Alg}} \sim \frac{\log d}{d} n.$$

Theorem 6

[G & Sudan 2014] The NAE-K-SAT problem exhibits the overlap gap property approximately at the "failure" point of the simple greedy algorithm. As a result no local algorithm can find a satisfying assignment above this threshold.

Remarks and Ongoing Work

Remarks and Ongoing Work

- Overlaps of $m > 2$ matrices should push the phase transition down from $\frac{5}{3}\sqrt{\frac{2}{3}}$ (work in progress).

Remarks and Ongoing Work

- Overlaps of $m > 2$ matrices should push the phase transition down from $\frac{5}{3}\sqrt{\frac{2}{3}}$ (work in progress).
- OGP in sparse regression (work in progress with Ilias Zadik).

Remarks and Ongoing Work

- Overlaps of $m > 2$ matrices should push the phase transition down from $\frac{5}{3}\sqrt{\frac{2}{3}}$ (work in progress).
- OGP in sparse regression (work in progress with Ilias Zadik).
- **Conjecture:** The Clique problem for $\mathbb{G}(n, p)$ exhibits an Overlap Gap Property at $\log_{\frac{1}{p}} n$ for general $m > 0$.

Remarks and Ongoing Work

- Overlaps of $m > 2$ matrices should push the phase transition down from $\frac{5}{3}\sqrt{\frac{2}{3}}$ (work in progress).
- OGP in sparse regression (work in progress with Ilias Zadik).
- **Conjecture:** The Clique problem for $\mathbb{G}(n,p)$ exhibits an Overlap Gap Property at $\log_{\frac{1}{p}} n$ for general $m > 0$.
- **Challenge:**

Random Constraint Satisfaction problem is tractable iff it does not exhibit the OGP.