Subgraph counting for planted cliques

Maryam Bahrani, Hrishikesh Khandeparkar

COS 521 - Final Project
Jan 17, 2017
Property Searching

• Variants: HAMPATH, EULERPATH, etc. *Varying complexities!*
Property Searching

• Variants: HAMPATH, EULERPATH, etc. *Varying complexities!*

• Our focus: MAXCLIQUE
  • Finding the largest clique in a graph
  • But is well known to be NP-hard
Property Searching

• Variants: HAMPATH, EULERPATH, etc. Varying complexities!
• Our focus: MAXCLIQUE (with relaxations)
  • Consider average case complexity
Property Searching

• Variants: HAMPATH, EULERPATH, *etc.* *Varying complexities!*
• Our focus: MAXCLIQUE (*with relaxations*)
  • Consider average case complexity
• Find MAXCLIQUE in graph from $G(n, 1/2)$
  • Properties well known – max-clique is $2 \log(n)$ with high probability.
Property Searching

• Variants: HAMPATH, EULERPATH, etc. Varying complexities!

• Our focus: MAXCLIQUE (with relaxations)
  • Consider average case complexity

• Find MAXCLIQUE in graph from \( G(n, 1/2) \)
  • Properties well known – max-clique is \( 2 \log(n) \) with high probability.
  • Recovering max clique is \( O(n \log n) \)
  • Recovering a clique of size \( \log(n) \) can be found greedily in poly-time
  • Still pretty bad – maybe make things easier by artificially planting a larger clique randomly?
A popular relaxation – planted cliques.

• Start with Erdös-Rényi random graph from $\mathbb{G}(n, 1/2)$
A popular relaxation – planted cliques.

- Start with Erdős-Rényi random graph from $\mathbb{G}(n, 1/2)$
- “Plant” a clique on $k$ random vertices by adding all the edges between them to get $\mathbb{G}^{(k)}(n, 1/2)$
A popular relaxation – planted cliques.

• Start with Erdös-Rényi random graph from $\mathbb{G}(n, 1/2)$

• “Plant” a clique on $k$ random vertices by adding all the edges between them to get $\mathbb{G}^{(k)}(n, 1/2)$

• Detection version:
  • Given a graph $G$, decide whether it was drawn from $\mathbb{G}(n, 1/2)$ or $\mathbb{G}^{(k)}(n, 1/2)$
A popular relaxation – planted cliques.

• Start with Erdös-Rényi random graph from $\mathbb{G}(n, 1/2)$

• “Plant” a clique on $k$ random vertices by adding all the edges between them to get $\mathbb{G}(k)(n, 1/2)$

• Detection version:
  • Given a graph $G$, decide whether it was drawn from $\mathbb{G}(n, 1/2)$ or $\mathbb{G}(k)(n, 1/2)$

• Recovery version:
  • Detection + if from $\mathbb{G}(k)(n, 1/2)$, identify the $k$ vertices on which the clique was planted
A popular relaxation – planted cliques.

- Start with Erdös-Rényi random graph from $\mathbb{G}(n, 1/2)$
- “Plant” a clique on $\kappa$ random vertices by adding all the edges between them to get $\mathbb{G}^{(\kappa)}(n, 1/2)$

- Detection version:
  - Given a graph $G$, decide whether it was drawn from $\mathbb{G}(n, 1/2)$ or $\mathbb{G}^{(\kappa)}(n, 1/2)$

- Recovery version:
  - Detection + if from $\mathbb{G}^{(\kappa)}(n, 1/2)$, identify the $\kappa$ vertices on which the clique was planted

Reduces to
A popular relaxation – planted cliques.

- Start with Erdös-Rényi random graph from $\mathbb{G}(n, 1/2)$
- “Plant” a clique on $k$ random vertices by adding all the edges between them to get $\mathbb{G}^{(k)}(n, 1/2)$

- **Detection version:**
  - Given a graph $G$, decide whether it was drawn from $\mathbb{G}(n, 1/2)$ or $\mathbb{G}^{(k)}(n, 1/2)$

- **Recovery version:**
  - Detection + if from $\mathbb{G}^{(k)}(n, 1/2)$, identify the $k$ vertices on which the clique was planted
  - Detection gets easier as $k$ grows

Reduces to
Known Algorithms

• Homework 3!
Known Algorithms

- Alon, Krivelevich, Sudakov (1998)
  - “Finding a large hidden clique in a random graph”
Known Algorithms

• Alon, Krivelevich, Sudakov (1998)
  • “Finding a large hidden clique in a random graph”
  • Use second largest eigenvalue of adjacency matrix for detection
Known Algorithms

• Alon, Krivelevich, Sudakov (1998)
  • “Finding a large hidden clique in a random graph”
  • Use second largest eigenvalue of adjacency matrix for detection
  • Works when planted clique is of size $k > 2.5\sqrt{n}$
Known Algorithms

- Alon, Krivelevich, Sudakov (1998)
  - “Finding a large hidden clique in a random graph”
  - Use second largest eigenvalue of adjacency matrix for detection
  - Works when planted clique is of size $k > 2.5\sqrt{n}$
  - Nifty trick also helps us recover.
Known Algorithms

- Alon, Krivelevich, Sudakov (1998)
  - “Finding a large hidden clique in a random graph”
  - Use second largest eigenvalue of adjacency matrix for detection
  - Works when planted clique is of size $k > 2.5\sqrt{n}$
  - Nifty trick also helps us recover.
  - Surprisingly, still is the best known algorithm that is polynomial
Known Algorithms

• Alon, Krivelevich, Sudakov (1998)
  • “Finding a large hidden clique in a random graph”
  • Use second largest eigenvalue of adjacency matrix for detection
  • Works when planted clique is of size $k > 2.5\sqrt{n}$
  • Nifty trick also helps us recover.
  • Surprisingly, still is the best known algorithm that is polynomial

• But largest clique in $G(n, 1/2)$ is $\sim 2 \log(n)$. Can we decrease this gap?
Can we do better?

• What about cliques of size \( o(\sqrt{n}) \)?
Can we do better?

- What about cliques of size $o(\sqrt{n})$?
  - Do spectral methods work?
    - Considering top eigenvalue of certain matrices
Can we do better?

• What about cliques of size $o(\sqrt{n})$?
  • Do spectral methods work?
  • Alon et al. algorithm is spectral method on adjacency matrix.
Can we do better?

• What about cliques of size $o(\sqrt{n})$?
  • Do spectral methods work?
  • Alon et al. algorithm is spectral method on adjacency matrix.
• How about spectral methods on other matrices, whose entries are polynomial in edge indicator variables?
Can we do better?

• What about cliques of size $o(\sqrt{n})$?
  • Do spectral methods work?
  • Alon et al. algorithm is spectral method on adjacency matrix.
• How about spectral methods on other matrices, whose entries are polynomial in edge indicator variables?
• How about generalizing spectral methods?
Can we do better?

• What about cliques of size $o(\sqrt{n})$?
  • Do spectral methods work?
  • Alon et al. algorithm is spectral method on adjacency matrix.
• How about spectral methods on other matrices, whose entries are polynomial in edge indicator variables?
• How about generalizing spectral methods?
  • Largest eigenvalue is approximately a polynomial in entries of matrix
Can we do better?

• What about cliques of size $o(\sqrt{n})$?
  • Do spectral methods work?
  • Alon et al. algorithm is spectral method on *adjacency matrix*.
• How about spectral methods on *other matrices*, whose entries are polynomial in edge indicator variables?
• How about generalizing spectral methods?
  • Largest eigenvalue is approximately a *polynomial* in entries of matrix
  • Spectral methods are special cases of SDP
Can we do better?

• What about cliques of size $o(\sqrt{n})$?
  • Do spectral methods work?
  • Alon et al. algorithm is spectral method on adjacency matrix.
• How about spectral methods on other matrices, whose entries are polynomial in edge indicator variables?
• How about generalizing spectral methods?
  • Largest eigenvalue is approximately a polynomial in entries of matrix
  • Spectral methods are special cases of SDP
• Not sure about relationship between polynomials and SDPs
Polynomials in edge indicators

• More powerful generalization of spectral methods
Polynomials in edge indicators

- More powerful generalization of spectral methods
- Interested in low degree polynomials
  - Expect that they fail when the planted cliques of size $o(\sqrt{n})$
Polynomials in edge indicators

• More powerful generalization of spectral methods
• Interested in low degree polynomials
  • Expect that they fail when the planted cliques of size $o(\sqrt{n})$
• Difficult to prove *all* low-degree polynomials fail
Polynomials in edge indicators

• More powerful generalization of spectral methods
• Interested in low degree polynomials
  • Expect that they fail when the planted cliques of size $o(\sqrt{n})$
• Difficult to prove *all* low-degree polynomials fail
  • Rule out a class of polynomials instead
Polynomials in edge indicators

• More powerful generalization of spectral methods
• Interested in low degree polynomials
  • Expect that they fail when the planted cliques of size $o(\sqrt{n})$
• Difficult to prove *all* low-degree polynomials fail
  • Rule out a class of polynomials instead

Observation: Counting $2\log(n)$-sized cliques works w.h.p.
(but takes exponential time, so we need to count smaller subgraphs)
Polynomials in edge indicators

• More powerful generalization of spectral methods
• Interested in low degree polynomials
  • Expect that they fail when the planted cliques of size $o(\sqrt{n})$
• Difficult to prove all low-degree polynomials fail
  • Rule out a class of polynomials instead
• Subgraphs can be written as monomials in edge indicators.
  • Subgraph counts are polynomials in edge indicators.
Polynomials in edge indicators

• More powerful generalization of spectral methods
• Interested in low degree polynomials
  • Expect that they fail when the planted cliques of size $o(\sqrt{n})$
• Difficult to prove all low-degree polynomials fail
  • Rule out a class of polynomials instead
• Subgraphs can be written as monomials in edge indicators.
  • Subgraph counts are polynomials in edge indicators.
• We will try ruling out subgraph count polynomials.
Subgraph Counting

• Detection Algorithm:
  • Input graph $G$
  • Compute $X$, the number of (non-induced) occurrences of a subgraph $H$
  • If $X > \text{threshold}$:
    • Output $\mathcal{G}_r(k)(n, 1/2)(\text{planted})$
  • Otherwise:
    • Output $\mathcal{G}(n, 1/2)(\text{random})$
Subgraph Counting

• Detection Algorithm:
  • Input graph $G$
  • Compute $X$, the number of (non-induced) occurrences of a subgraph $H$
  • If $X > \text{threshold}$:
    • Output $\mathbb{G}^{(k)}(n, 1/2)$ (planted)
  • Otherwise:
    • Output $\mathbb{G}(n, 1/2)$ (random)

Constant sized subgraphs admit polynomial algorithms
Subgraph Counting

• Detection Algorithm:
  • Input graph $G$
  • Compute $X$, the number of (non-induced) occurrences of a subgraph $H$
  • If $X > \text{threshold}$:
    • Output $G^{(k)}(n, 1/2)\text{(planted)}$
  • Otherwise:
    • Output $G(n, 1/2)\text{(random)}$

• When will this algorithm likely succeed?
  • If planting a clique changes the count of $H$ by more than the standard deviation of the count of $H$ in a random graph
Subgraph Counting

\( H := \text{subgraph with } v \text{ vertices and } f \text{ edges} \)

\( X := \text{number of (non-induced) occurrences of } H \)

\( \mu := \mathbb{E}_{G(n,1/2)}[X] \)

\( \mu^{(k)} := \mathbb{E}_{G^{(k)}(n,1/2)}[X] \)

\( \sigma := \sqrt{\text{Var}[X]} \)

We can detect only if \( \mu^{(k)} - \mu \geq \sigma \).
Our Results

1. When the planted clique size is $k = o(\sqrt{n})$:
   - Counting **constant-sized** subgraphs does not help with detection
   - Rules out poly-time subgraph count algorithms
Our Results

1. When the planted clique size is $k = o(\sqrt{n})$:
   • Counting \textbf{constant-sized} subgraphs \textbf{does not} help with detection
     • Rules out poly-time subgraph count algorithms

2. When the planted clique size is $k = O(\sqrt{\frac{n}{\log(n)}})$:
   • Counting $o(\log(n))$ \textbf{- sized} cliques \textbf{does not} help with detection
     • Rules out $n^{o(\log(n))}$- time clique count algorithms
Our Results

1. When the planted clique size is $k = o(\sqrt{n})$:
   • Counting constant-sized subgraphs does not help with detection
     • Rules out poly-time subgraph count algorithms

2. When the planted clique size is $k = O(\sqrt{\frac{n}{\log(n)}})$:
   • Counting $o(\log(n))$-sized cliques does not help with detection
     • Rules out $n^{o(\log(n))}$-time clique count algorithms

Shown for cliques, conjectured for any subgraph
Sketch (upper-bounding $\mu^{(k)} - \mu$ for $p = \frac{1}{2}$)
Sketch (upper-bounding $\mu^{(k)} - \mu$ for $p = \frac{1}{2}$)

Number of ways of putting $H$ on $\nu$ vertices

$$\mu = \mathbb{E}_{G(n,1/2)}[X] = \binom{n}{\nu} \cdot C_H \cdot \frac{1}{2^f}$$
Sketch (upper-bounding $\mu^{(k)} - \mu$ for $p = \frac{1}{2}$)

Number of ways of putting $H$ on $v$ vertices

$$\mu = \mathbb{E}_{G(n, 1/2)}[X] = \binom{n}{v} \cdot C_H \cdot \frac{1}{2^f}$$

$$\mu^{(k)} = \mathbb{E}_{G^{(k)}(n, 1/2)}[X] \leq \left[ \binom{n-k}{v} + \binom{n-k}{v-1} \cdot \binom{k}{1} + \sum_{l=2}^{v} \binom{n-k}{v-l} \binom{k}{l} 2^{(l/2)} \right] \cdot C_H \cdot \frac{1}{2^f}$$

- no vertex from planted clique
- one vertex from planted clique
- at most $\binom{l}{2}$ edges provided by planted clique
- $l$ vertices from planted clique
Sketch (upper-bounding $\mu^{(k)} - \mu$ for $p = \frac{1}{2}$)

Number of ways of putting $H$ on $\nu$ vertices

$$\mu = \mathbb{E}_{G(n, 1/2)}[X] = \binom{n}{\nu} \cdot C_H \cdot \frac{1}{2^f}$$

$$\mu^{(k)} = \mathbb{E}_{G^{(k)}(n, 1/2)}[X] \leq \left[ \binom{n-k}{\nu} + \binom{n-k}{\nu-1} \binom{k}{1} + \sum_{l=2}^{\nu} \binom{n-k}{\nu-l} \binom{k}{l} 2\binom{l}{2} \right] \cdot C_H \cdot \frac{1}{2^f}$$

- **no vertex from planted clique**
- **one vertex from planted clique**
- **$l$ vertices from planted clique**

$$\sum_{i=0}^{\nu} \binom{n-k}{\nu-l} \binom{k}{l} = \binom{n}{\nu}$$

$$= \binom{n}{\nu} \cdot C_H \cdot \frac{1}{2^f} + \sum_{l=2}^{\nu} \binom{n-k}{\nu-l} \binom{k}{l} \cdot C_H \cdot \frac{2\binom{l}{2} - 1}{2^f}$$
Sketch (upper-bounding $\mu^{(k)} - \mu$ for $p = \frac{1}{2}$)

Number of ways of putting $H$ on $v$ vertices

$$\mu = \mathbb{E}_{G(n,1/2)}[X] = \binom{n}{v} \cdot C_H \cdot \frac{1}{2^f}$$

$$\mu^{(k)} = \mathbb{E}_{G^{(k)}(n,1/2)}[X] \leq \left[ \binom{n-k}{v} + \binom{n-k}{v-1} \binom{k}{1} + \sum_{l=2}^{v} \binom{n-k}{v-l} \binom{k}{l} 2^{\binom{l}{2}} \right] \cdot C_H \cdot \frac{1}{2^f}$$

$$= \binom{n}{v} \cdot C_H \cdot \frac{1}{2^f} + \sum_{l=2}^{v} \binom{n-k}{v-l} \binom{k}{l} \cdot C_H \cdot \frac{2^{\binom{l}{2}} - 1}{2^f}$$

no vertex from planted clique

one vertex from planted clique

$l$ vertices from planted clique

at most $\binom{l}{2}$ edges provided by planted clique
Sketch (upper-bounding $\mu^{(k)} - \mu$ for $p = \frac{1}{2}$)

- Constant-sized subgraphs

$$\mu^{(k)} - \mu = O \left( \binom{n - k}{v - 2} \binom{k}{2} \right)$$

- $o(\log(n))$-sized cliques

$$\mu^{(k)} - \mu = O \left( v \cdot \binom{n - k}{v - 2} \binom{k}{2} \frac{2^2}{2f} \right)$$
Sketch (lower-bounding $\sigma$)

$$\sigma^2 = \Omega \left( \binom{n}{v} \binom{v}{2} \binom{n-v}{v-2} \cdot \frac{C_H^{(2)}}{2^{2f}} \right)$$

number of ways of putting two copies of $H$ on $2v - 2$ vertices, with exactly 2 fixed vertices and 1 edge in common

for constant-sized subgraphs and $o(\log(n))$ cliques
Sketch (lower-bounding $\sigma$)

$$\sigma^2 = \Omega \left( \binom{n}{v} \binom{v}{2} \binom{n-v}{v-2} \cdot \frac{C^{(2)}_H}{2^2f} \right)$$

• Key ideas:
  • Similar to homework 1, except need *lower* bound
  • Need to worry about negative terms

number of ways of putting two copies of $H$ on $2v - 2$ vertices, with exactly 2 fixed vertices and 1 edge in common

for constant-sized subgraphs and $o(\log(n))$ cliques
Sketch (lower-bounding $\sigma$)

$$\sigma^2 = \Omega \left( \binom{n}{v} \binom{v}{2} \binom{n - v}{v - 2} \cdot \frac{C_H^{(2)}}{2^2f} \right)$$

- Key ideas:
  - Similar to homework 1, except need lower bound
    - Need to worry about negative terms
  - Decreasing series, only first term asymptotically significant
  - All terms can’t cancel out for constant subgraphs

number of ways of putting two copies of $H$ on $2v - 2$ vertices, with exactly 2 fixed vertices and 1 edge in common

for constant-sized subgraphs and $o(\log(n))$ cliques
Sketch (lower-bounding $\sigma$)

$$\sigma^2 = \Omega\left( \binom{n}{v} \binom{v}{2} \binom{n - v}{v - 2} \cdot \frac{C_H^{(2)}}{2^{2f}} \right)$$

- Key ideas:
  - Similar to homework 1, except need *lower* bound
    - Need to worry about negative terms
  - Decreasing series, only first term asymptotically significant
  - All terms can’t cancel out for constant subgraphs
  - For general subgraphs, $C_H^{(2)} = \Omega(1)$
    - can cause problems for $o(\log(n))$-sized subgraphs

number of ways of putting two copies of $H$ on $2v - 2$ vertices, with exactly 2 fixed vertices and 1 edge in common

for constant-sized subgraphs and $o(\log(n))$-cliques
Sketch (putting it together)

• For constant-sized subgraphs and $o(\log(n))$-cliques: $\mu^{(k)} - \mu = o(\sigma)$

• Such deviations are likely in a random graph
  • Detection impossible by counting subgraphs.
Conclusion and Future Directions

• Generalize to all polynomials in edge indicators

• Explore relationship between SDPs and polynomials

• Consider other regimes, e.g. vanishing $p$