NATIONAL TAIWAN UNIVERSITY

COLLEGE OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Undergraduate Research Report

Computational Tradeoff in Network Problems

Student:
Chi-Ning Chou

Advisor:
I-Hsiang Wang

January 25, 2016

Abstract

This semester I focused on the computational tradeoff in network problems. Concretely, I've studied how error, algorithm time, and problem size affect each other in hidden clique/subgraph problems. These problems are highly related to applications in Biology, Economics, Cryptography, Statistics, etc. Once we fully understand the computational tradeoff of hidden clique/subgraph problems, people can have a complete view about the practical issues of these related applications and thus making progress in their own researches.

In the first month of this semester, I surveyed papers about the computational issues in hidden submatrix and hidden clique problems. There are several approaches to formulate the tradeoff. Basically, people fixed two of the three factors and see how the other behaves. The goal here is to characterize which regions are achievable and which are not. To show achievability, it suffices to present an algorithm. However, to assert non-achievability, currently there's no unified method to do so. Details about this survey are provided in Appendix A.

In the next two months, I surveyed one of the powerful computational lower bound techniques: sum of square (SOS) method. SOS provides us a hierarchical lower bound for polynomial time algorithm of certain problem. Take hidden clique problem for example, Meka, Potechin, and Wigderson in [MPW15] showed that one cannot have a $O(n^d)$ time algorithm for hidden clique problem with clique size $O((\frac{\sqrt{n}}{\log n})^{1/d}/C^d)$, where n is the number of vertices and C is some constant. SOS method captures the power of various algorithm involving semidefinite programming. As a result, although SOS is not an universal lower bound method, it still gives us great confidence about the computational hardness of certain problem. Details about this survey are provided in Appendix B.

In the last month of this semester, I tried to worked on some extended problems of hidden subgraph problem. I deduced information-theoretical limit of locating problem and distinguishing two distinct subgraphs problem. After weeks of trail and error, I found out that it is nontrivial to deal with general subgraph. Thus, I did some surveys on the applications of these problems and tried to find some interesting structure so that I can focus on certain category of graphs and hopefully make some progress. Details about my current works are provided in Appendix C.

This report will focus on the hidden clique/subgraph problems. In Section 1, motivations and applications of these problems in various fields will be discussed. Then, historical works and current status of related problems will be provided in Section 2. I would formulate the hidden subgraph problem and show some information-theoretical results in Section 3. Finally, conclusions and future works will be drawn in Section 4.

Contents

1 Motivations

Hidden clique/subgraph problems asked the following questions:

Problem

Given a planted/unplanted graph, can you tell whether it is planted or not? If so, can you find where it is?

Clearly, this directly relates to the problems that involving finding hidden structure in a large network. For instance, given a social network, can one find a hidden community efficiently? On the other hand, for theoretical reason, planted clique problem is also a important average-case complexity problem which can be a nice hardness assumption. Once we understand the underlying computational tradeoff of the planted clique/subgraph problems, the results can be immediately extended to lots of applications. In the rest of this section, I would present some applications that are related to our interested network problems.

1.1 Finding subtle signal in DNA sequences

In [PS$^+$00], Pevzner and Sze presented an combinatorial algorithm for finding subtle signal in DNA sequences. The problem they were facing is informally stated as follow

Problem

Find an unknown signal of length 15 with 4 mismatches in a sample of sequences which is 600 nucleotides long.

Generally speaking, this problem can be extended to finding a length l signal/pattern in a huge sequences consisting of finite letters with error-tolerances d. Suppose we have t i.i.d. length n samples $S = \{s_1, ..., s_t\}$, we construct a graph $G(S,l,d) = (V,E)$ as follow

$$V = \{s_{ij} | i = 1, ..., t, j = 1, ..., n-l+1\}$$
$$E = \{(s_{ij}, s_{pq}) | i \neq j, |s_{ij} - s_{pq}| \leq d\}$$

Intuitively, the vertex set of $G(S,l,d)$ collects all length l substrings in S and two substrings in distinct samples has an edge if the hamming distortion among them are less than or equal to d. With this construction, one can see that a $k-$clique in $G(S,l,d)$ corresponds to a length l signal with at most $2d$ mismatches. As a result, we can reduce the finding subtle signal problem to the hidden clique problem. As long as we have efficient algorithm for hidden clique problem, we can have a nice algorithm to find hidden pattern in DNA sequences.

1.2 Hardness of certifying restricted isometry property (RIP)

Restricted isometry property is an important in compressed sensing because once the sensing matrix enjoys RIP, we can have a nice algorithm to detect the hidden sparse structure in a signal. Formally, we say a sensing matrix Φ has RIP of order k with parameter δ if for any $k-$sparse vector x, we have

$$(1-\delta)||x||^2 \leq ||\Phi x||^2 \leq (1+\delta)||x||^2$$

In [KZ14], Koiran and Zouzias reduced hidden clique problem to certifying whether a matrix satisfying RIP. Concretely, given a graph $G=(V,E)$, let matrix $C(G)$ to be the Cholesky decomposition of $I_n + cA/\sqrt{n}$, where n is the number of vertices in G, c is some absolute constant, and A is the adjacency matrix of G where $A_{ii}=0$, $A_{ij}=1$ if $(i,j)\in E$ and $A_{ij}=-1$ if $(i,j)\notin E$ for $i\neq j$.

After doing some linear algebraic reasoning, one can find out that $C(G)$ has RIP iff G does not have a k clique. As a result, if one can efficiently certify the RIP of arbitrary matrix, then he/she can efficiently solve the hidden clique problem. Since we assume hidden clique problem is hard in certain region $k=O(n^{1/2-\delta})$ for any $\delta>0$, the hardness of certifying RIP is guaranteed by the hardness of hidden clique problem.

1.3 Hardness of sparse PCA

Berthet and Rigollet asserted the hardness of principal component detection via hidden clique problem in [BR13]. Let's formulate the problem as follow. Suppose there are two hypotheses

$$\begin{aligned}\mathscr{H}_0 &= X \sim N(0,I_d)\\ \mathscr{H}_1 &= X \sim N(0,I_d+\theta vv^T),\ v\in B_0(k)\end{aligned}$$

the null hypothesis is the standard d dimensional gaussian distribution while the alternative hypothesis add an unknown $k-$sparse noise. The goal is to determine which distribution is the given graph from. There's an optimal test for the region $\theta=\Omega(\sqrt{\frac{k\log d}{n}})$ via sparse eigenvector statistics. However, to compute this statistics is NP-hard and the best polynomial-time algorithm can only serve for the region $\theta=\Omega(\sqrt{\frac{k^2\log d}{n}})$.

As a result, Berthet and Rigollet showed the impossibility of having polynomial-time algorithm for $\theta = \Omega(\sqrt{\frac{k^\alpha \log d}{n}})$ with $\alpha < 2$ in the sense that once the algorithm does exist, there will be a polynomial time algorithm for planted clique problem with $k = O(n^{1/2-\delta})$ for some $\delta > 0$, which is conjectured to be impossible.

1.4 Hardness of approximating Nash equilibrium

In algorithmic game theory, a major open problem is the quest for a PTAS (polynomial-time approximation scheme) for Nash equilibrium in a two-player game. Namely, whether there's a family of polynomial-time algorithms to approximate Nash equilibrium with any error ε. The status of the existence of PTAS is unknown, and there are positive and negative results suggesting the existence/non-existence. For the positive part, Chen, Deng, and Teng showed that there are a FPTAS (fully polynomial-time approximation scheme) for the problem. That is, there's a family of polynomial-time algorithms in both problem size n and the reciprocal of error $1/\varepsilon$ for the problem. However, for the negative part, a similar problem of maximizing the social welfare is known to be NP-hard.

Hazan and Krauthgamer in [HK11] reduced the planted clique problem to two-player Nash equilibrium. Concretely, the problem can be formulated as follow: There are two players, one is Row, and the other is Column. Both of them can make a choice among n strategies. As we allow mixed strategy, a single strategy can be viewed as a length n probability vector. Now, suppose Row chooses single strategy $i \in [n]$ and Column chooses single strategy $j \in [n]$, the payoffs for each them is $R(i,j)$ and $C(i,j)$ respectively. Namely, matrix R, C record the payoffs and once Row chooses mixed strategy x and Column chooses mixed strategy y, the payoffs are $x^T Ry$ and $x^T Cy$ respectively.

The goal for finding an $\varepsilon-$equilibrium is to find a pair (x,y) such that for all mixed strategies $(\tilde{x},\tilde{y})$ we have

$$\tilde{x}^T R\tilde{y} \leq x^T Ry + \varepsilon$$
$$\tilde{x}^T C\tilde{y} \leq x^T Cy + \varepsilon$$

Finally, the reduction in [HK11] showed that once we have a polynomial time algorithm for any ε (*i.e.,* PTAS), then there's polynomial time algorithm for any clique with size $c\log n$. The idea of reduction is quite straightforward, we append some random noise to the adjacency matrix of the given graph so that if the graph has a k clique, then there's an equilibrium with value 1 w.h.p. On the other hand, if we have an $\varepsilon-$approximate algorithm with value $\geq 1-\varepsilon$, then we can find a clique in the graph.

1.5 Hardness assumption for public-key encryption

In the world of modern cryptography, people use (conjectured) hard problems to reduce to cryptographic construction so that the security is guaranteed by the hardness of these problems such as discrete logarithmic problem, lattices problems etc. However, there are two important concerns for finding a good hard problem for reduction: hard enough, and easy to be reduced. Most of the time these two criterion cannot be satisfied simultaneously, thus, though lots of cryptographic constructions have been built, people are still working on finding new hardness assumption.

In [ABW10], Applebaum, Barak, and Wigderson showed how to reduce lots of known/conjectured hard problem to public-key cryptography (PKE). Basically, they characterized these problems into three hardness assumptions: dLIN, DUE, and DSF. Our hidden clique problem is categorized as a DUE assumption, which is a decision problem for distinguishing a normal random graph and a random graph with some planted dense structure.

The details of reducing DUE to PKE is omitted here since it's not really the point here. To me, the message from this research is that the hardness of hidden clique problem can actually been weaken to DUE problem. In this report, I would like to write down the definition of DUE problem and maybe I will study the details later.

Problem

A DUE(m,q,d) problem is to distinguish

- A random bipartite graph with m vertices on left and n vertices on right where the degree of each vertex is d.
- A random bipartite graph as previous plus randomly choose $q-$sized subset S of left vertices and ensure S will only have $q/3$ neighbors.

2 Historical works

The hidden clique problem is an important average-case complexity problem in theoretical computer science. It can be used as the hardness assumption for approximation algorithm and various applications mentioned in Section 1. In this section, I would present historical works on this line of works and show some current status of the problem.

2.1 History

For the convenience of reading, here I list the history of hidden clique problem instead of putting everything in a single paragraph.

- **1984**: Karp suggested that finding clique with size $k = (1+\varepsilon)\log n$ is hard. [Kar84]
- **1995**: Kucera formulated the hidden clique (a.k.a. planted clique) problem. [Kuč95]
- **1998**: Alon, Krivelevich, and Sudakov presented a polynomial algorithm for $k = \Omega(\sqrt{n})$. [AKS98]
- **2002**: Feige, and Krauthgamer showed degree d Lovasz-Schrijver lower bound for $k = O(\sqrt{n/2^d})$.
- **2015**: Meka, Potechin, and Wigderson showed sum of square lower bound for $k = O((\sqrt{n}/\log n)^{1/d}/C^d)$. [MPW15]

The message here is that in general people believe that there's no polynomial-time algorithm for hidden clique problem with clique size $k = O(n^{1/2-\varepsilon})$ for any $\varepsilon > 0$. However, there's no unified lower bound results been discovered. What we only have is some conditional lower bound such as Lovasz-Schrijver and SOS hierarchy.

2.2 Current status of hidden clique problem

Information-theoretically, one can use a quasi-polynomial time algorithm to detect hidden clique with size $k = \Omega(\log n)$ because with some probabilistic argument, one can show that the expected largest clique in a ER-random graph is $2\log n$. However, current best polynomial-time algorithm can only detect hidden clique with size $k = \Omega(\sqrt{n})$. That is, there exists gap between best polynomial-time algorithm and optimal algorithm.

Due to the huge computational gap in hidden clique problem, people began working on constructing the clique size lower bound for polynomial-time algorithm. However, there's no unified lower bound for this problem. Namely, people can only work on conditional hardness proof such as the sum of square method discussed in Appendix B. Other than semidefinite programming lower bound, there's also an interesting line of works focusing on lower bound for the so called statistical algorithm. In [FGR$^+$13], Feldman et al. showed that by lower bounding the number of statistical queries, one can show $k = O(n^{1/2-\varepsilon})$ lower bound for a similar problem: hidden bipartite clique problem. To sum up, please refer to Figure 1 for graphical summary.

Basically, current focusing on the complexity issue of hidden clique problem is on the lower bound of matching the known optimal polynomial-time algorithm for clique size

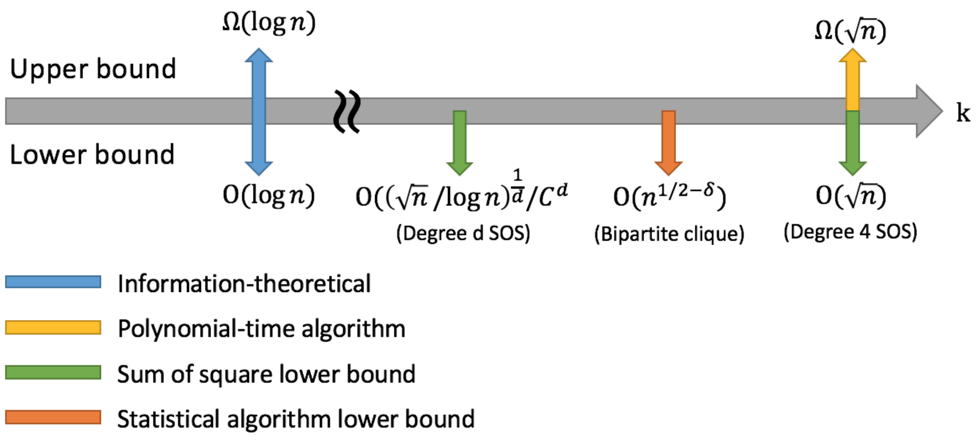

Figure 1: Current status of hidden clique problem.

$k = \Omega(\sqrt{n})$.

2.3 Current status of hidden subgraph problem

In [JM15], Javadi and Montanari studied the general hidden subgraph problem. They presented information-theoretical results and polynomial-time algorithm. Their current status can be represented in Figure

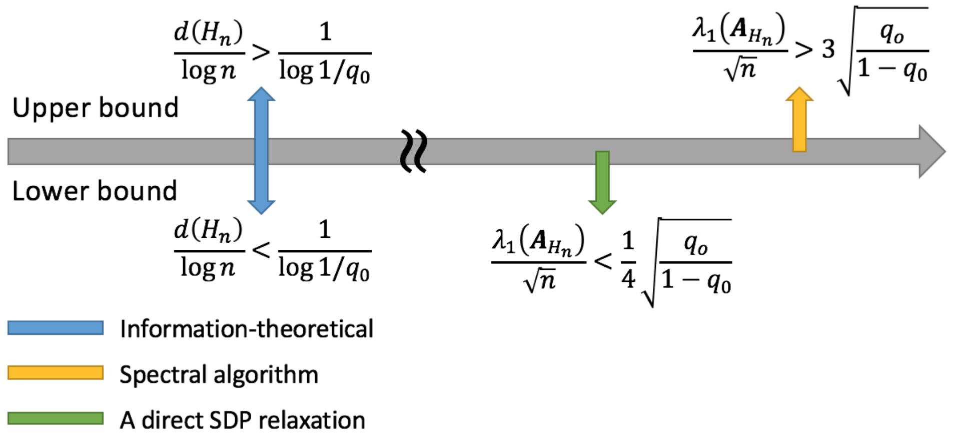
d(H_n)/log n > 1/log 1/q_0
λ_1(A_{H_n})/√n > 3 √(q_o/(1 − q_0))
Upper bound
Lower bound
d(H_n)/log n < 1/log 1/q_0
λ_1(A_{H_n})/√n < 1/4 √(q_o/(1 − q_0))
Information-theoretical
Spectral algorithm
A direct SDP relaxation

Figure 2: Current status of hidden subgraph problem.

The hidden subgraph problem is still a not-so-well studied research, the actual application and formal hardness construction remains widely open. In my opinion, it is possible to show that for certain type of subgraph, the results of hidden subgraph problem will be the same as the hidden clique problem.

3 Current results

Our goal here is to explore the difficulty of detecting/locating substructure in a network. We formalize the problems into a decision problem as whether a random graph is planted or not, and a search problem as where's the hidden substructure. The detecting problem was discussed by [JM15], which showed a threshold for the density of the substructure. As the density is greater than $\frac{\log n}{\log 1/q_0}$, then we can distinguish the planted network from a totally random network with connecting probability q_0. In this report, we tried to extend the result to locating problem. The results are summarized in Table 1. Moreover, we tried to distinguish two distinct dense substructures. Interestingly, we define a notion of the distortion among graphs in Definition 9 and showed a similar result in Theorem 11.

3.1 Locating subgraph

In this section, we consider the locating subgraph problem in a planted random graph. We derived sufficient condition and necessary condition to locate a hidden subgraph. Specifically, we showed that as the edge expansion $\alpha = \omega(\frac{\log n}{\log 1/q_0})$, the probability of locating error will be negligible, where n is the size of the graph and q_0 is the probability of random edges. And as $d_{max} = o(\frac{\log n}{\log 1/q_0})$, the probability of locating error will converge to 1, where d_{max} is the maximum degree in the subgraph.

3.1.1 Problem setup

We have a family of graphs $\{H_n\}$ which will be planted into a sequence of random graphs G_n indexed by the number of vertices with an index mapping $\phi_n^* : V(H_n) \to V(G_n)$. Note that this mapping is one-to-one. Besides the planted edges, other possible edges will have probability $0 < q_0 < 1$ to be added into G_n.

With the above scenario, we should formally define two things here: the probability space of G_n given a subgraph H_n and a index mapping ϕ_n^*, and the possible index mapping space.

Definition 1 (planted graph space) *Given a subgraph H_n and an index mapping ϕ_n^*, we define the probability space of planted graph as*

$$\mathbb{G}_{n,H_n,\phi_n^*} = \{G_n : |V(G_n)| = n,\ (\phi_n^*(i), \phi_n^*(j)) \in E(G_n)\ \forall (i,j) \in E(H_n)\}$$

, where the probability density of a graph $G_n \in \mathbb{G}_{n,H_n,\phi_n^*}$ is

$$\mathbb{P}_{n,H_n,\phi_n^*}[G_n] = q_0^{|E(G_n)|-|E(H_n)|} \cdot (1-q_0)^{\frac{n(n-1)}{2}-|E(G_n)|}$$

Definition 2 (index mapping space) *Given a subgraph H_n, the index mapping space is*

$$\mathbb{I}_{n,H_n} = \{\phi : \phi : V(H_n) \to [n]\}$$

An subgraph locating algorithm is a function $\psi_n : \mathbb{G}_{n,H_n,\phi_n^*} \to \mathbb{I}_{n,H_n}$ and we can directly define the locating error as $\mathbb{P}_{G_n \xleftarrow{R} \mathbb{G}_{n,H_n,\phi_n^*}}[\psi(G_n) \neq \phi_n^*]$.

3.1.2 Node-flipping error

We start our analysis from a special type of error: the *node-flipping error*. That is, $\psi_n(G_n)$ and ϕ_n^* differ in exactly one node.

Remark: Node-flipping error is only a *sufficient condition* for a locating error to happen. Note that there might be an cyclic error involved more than one vertices without having a node-flipping error.

Now, let's define events that indicate the node-flipping error:

- $R_{i,j} = \{\phi : \phi(j) = i\}$, where $i \in H_n, j \in G_n \backslash \phi_n^*(H_n)$. That is, mismatching node i in H_n to node j in G_n which is not a planted node. Define the single-to-single node-flipping error event as $\mathscr{E}_{i,j} = \{R_{i,j} \neq \emptyset\}$.
- $R_i = \cup_{j \in G_n \backslash \phi_n^*(H_n)} R_{i,j}$, which is the event that i in H_n is replaced by at least one of the non-planted nodes. Define the single node-flipping error as $\mathscr{E}_i = \{R_i \neq \emptyset\}$.
- $R = \cup_{i \in H_n}$, which is the event that at least one of a node in H_n is replaced by at least one of the non-planted nodes. Define the node-flipping error set as $\mathscr{E}^{node} = \{R \neq \emptyset\}$.

We have the following theorem.

Theorem 3 (node-flipping error) *Given H_n, ϕ_n^* and generate G_n from $\mathbb{G}_{n,H_n,\phi_n^*}$, we have*

- *For single-to-single node-flipping error:*

$$\mathbb{P}[\mathscr{E}_{i,j}] = q_0^{d_{H_n}(i)}$$

- *For single node-flipping error:*

$$\mathbb{P}[\mathscr{E}_i] = 1 - [1 - q_0^{d_{H_n}(i)}]^{n-|H_n|}$$

- *For node-flipping error:*

$$1-[\prod_{i\in H_n} 1-q_0^{d_{H_n}(i)}]^{n-|H_n|}e^{\Delta/(1-q_0)^2}\leq \mathbb{P}[\mathscr{E}^{node}]\leq 1-[\prod_{i\in H_n} 1-q_0^{d_{H_n}(i)}]^{n-|H_n|}$$

, where $\Delta=\sum_{i,i'\in H_n}\mathbf{1}_{R_i\sim R_{i'}}\leq d_{max}^2$*, and* $d_{max}=\max_{i\in H_n} d_{H_n}(i)$*. Especially,*

$$P[\mathscr{E}^{node}]\leq e^{|H_n|\cdot[\frac{d_{max}^2}{2(1-q_0)^2}-(n-|H_n|)\cdot q_0^{d_{max}}]}$$

Proof: See Appendix C.1. ■

Remark: Since node-flipping error is not a necessary condition for locating error, it only meaningful to consider the lower bound of node-flipping error, or equivalently, the upper bound of no node-flipping error.

Corollary 4 (necessary condition for node error) *If* $d_{max}=o(\frac{\log n}{\log 1/q_0})$*, we have*

$$\lim_{n\to\infty} P[\mathscr{E}^{node}]=1$$

Proof: Consider

$$\begin{aligned}\log\frac{(n-|H_n|)q_0^{d_{max}}}{d_{max}^2/2(1-q_0)^2}&=\log(n-|H_n|)-d_{max}\log 1/q_0-2\log d_{max}-\log 2(1-q_0)^2\\&>\log(n-|H_n|)-o(\log n)-2\log o(\frac{\log n}{\log 1/q_0})+\log 2(1-q_0)^2\\&=\Omega(\log n)\end{aligned}$$

That is, $\frac{d_{max}^2}{2(1-q_0)^2}=O(\frac{(n-|H_n|)q_0^{d_{max}}}{n})$ Thus, by Theorem 3, we have

$$\begin{aligned}P[\mathscr{E}^{node}]&\leq e^{|H_n|\cdot[\frac{d_{max}^2}{2(1-q_0)^2}-(n-|H_n|)\cdot q_0^{d_{max}}]}\\&\leq e^{|H_n|\cdot[\frac{1}{n}(n-|H_n|)\cdot q_0^{d_{max}}-(n-|H_n|)\cdot q_0^{d_{max}}]}\\&\xrightarrow{n\to\infty} 0\end{aligned}$$

■

3.1.3 Set-flipping error

Now, let's extend the idea in Section 3.1.2 to a more general setting: set-flipping error. That is, consider the error that misidentifying more than one nodes. Before we introduce the error event for set-flipping error, let's first define some useful notation:

- For $S \subseteq V(G)$, $\partial S := \{v \in V(G) \backslash S : \exists u \in S \; s.t. \; (u,v) \in E(G)\}$, which is the border of vertices set S.
- Overload the index mapping over vertices set. For $S \subseteq V(G)$, define $\phi(S) := \{\phi(v) : v \in S\}$.
- Recall the vertex expansion $\alpha := \min_{S \subseteq V(G)} \frac{|\partial S|}{|S|}$.

We define the following three types of error event for set-flipping error:

- For $S \in V(H_n)$ and $S' \in V(G_n \backslash H_n)$, define $R_{SS'} := \{\phi : \phi(S') = S\}$ to be the set-flipping event of S to S'.
- For $S \in V(H_n)$, define $R_S := \cup_{S' \subseteq V(G_n \backslash H_n)} R_{SS'}$ to be the set-flipping event of S.
- For integer $1 \leq k \leq |H_n|$, define $R_k := \cup_{S \in V(H_n):|S|=k} R_S$ to be the set-flipping event of size k.
- Define $R^e := \cup_{1 \leq k \leq |H_n|} R_k$ to be the set-slipping event.

With the above notion, we have a immediate upper bound for each set-flipping event.

Theorem 5 (upper bound for set-flipping error) *Given H_n, ϕ_n^* and generate G_n from $\mathbb{G}_{n,H_n,\phi_n^*}$, we have*

- *For single-to-single set-flipping error*
$$P[R_{SS'} \neq \emptyset] \leq q_0^{\alpha|S|} |S|!$$
- *For single set-flipping error*
$$P[R_S \neq \emptyset] \leq n^{|S|} q_0^{\alpha|S|}$$
- *For set-size set-flipping error*
$$P[R_k \neq \emptyset] \leq \frac{(n \cdot |H_n| \cdot q_0^{\alpha})^k}{k!}$$
- *For set-flipping error*
$$P[R^e \neq \emptyset] \leq e^{n \cdot |H_n| \cdot q_0^{\alpha}} - 1$$

Proof: See Appendix C.2. ■

Immediately, given an error level ε, we get the following sufficient condition to achieve:

Corollary 6 (sufficient condition for no error) *Given $\varepsilon > 0$, if $\alpha \geq \frac{\log n \cdot |H_n|/\varepsilon}{\log 1/q_0}$, then*

$$P[R^e] \leq \varepsilon + o(\varepsilon)$$

That is, if $\alpha = \omega(\frac{\log n \cdot |H_n|}{\log 1/q_0}) = \omega(\frac{\log n}{\log 1/q_0})$

$$\lim_{n\to\infty} P[\textit{node-flipping error}] = 0$$

Proof: Form Theorem 5, we have

$$\begin{aligned} P[R^e] &\leq e^{n\cdot|H_n|\cdot q_0^{\alpha}} - 1 \leq e^{n\cdot|H_n|\cdot q_0^{\frac{\log n\cdot|H_n|/\varepsilon}{\log 1/q_0}}} - 1 \\ &= e^{e^{\log n+\log|H_n|+(\frac{\log n+\log|H_n|-\log\varepsilon}{\log 1/q_0})\log q_0}} - 1 = e^{e^{\log\varepsilon}} - 1 \\ &= e^{\varepsilon} - 1 < \varepsilon + o(\varepsilon) \end{aligned}$$

■

3.1.4 Results

From Corollary 4 and Corollary 6, we have the necessary and sufficient condition for locating subgraph: That is, if $d_{max} = o(\frac{\log n}{\log 1/q_0})$, the probability of node-flipping error

Necessary	$d_{max} = o(\frac{\log n}{\log 1/q_0})$
Sufficient	$\alpha = \omega(\frac{\log n}{\log 1/q_0})$

Table 1: Resutls of locating subgraph.

to happen is non-negligible. And if $\alpha = \omega(\frac{\log n}{\log 1/q_0})$, the probability of locating error to happen is negligible.

3.2 Distinguish two distinct dense subgraph

Now, we consider the problem of distinguish two distinct dense subgraph. The motivation is that a network might consist a hidden substructure H_0 while in the meantime we want to detect whether the network contains another substructure H_1. Two structures H_0 and H_1 may have very different properties but as they are both dense and somehow being similar, detecting error might be non-negligible. In this section, we want to find the necessary and sufficient conditions for us to distinguish two different subgraph in a random planted setting.

3.2.1 Problem setup

Suppose there are two sequences of subgraph $\{H_{0,n}\}$ and $\{H_{1,n}\}$, where the number of vertices in each subgraph is $|H_{0,n}| = |H_{1,n}| = v_n$. They induce two families of distribution $\mathbb{G}_{0,n}$ and $\mathbb{G}_{1,n}$ such that the nth graphs contain n vertices and we plant $H_{0,n}$ and $H_{1,n}$ into the graph respectively while the rest edges are fired up with probability q_0. Since the definition of $\mathbb{G}_{0,n}$ and $\mathbb{G}_{1,n}$ are almost the same as Definition 1, we don't formally state their definition here.

To show the sufficient and necessary condition for distinguishability, we adopt the results in likelihood ratio test, which provided us a useful lemma. Now, we first derive the likelihood ratio of our problem then present the likelihood ratio test lemma.

Proposition 7 (likelihood ratio of two distinct subgraphs) *Given G with n vertices, the likelihood ratio of G following $\mathbb{G}_{0,n}$ or $\mathbb{G}_{1,n}$ is*

$$\mathscr{L}(G) = \frac{N(G,H_1)}{N(G,H_0)} \times q_0^{e(H_0)-e(H_1)}$$

, where $N(G,H_i)$ is the number of valid labeling for H_i in G and $e(H_i)$ is the number of edge in H_i.

Proof:

$$\begin{aligned}\mathscr{L}(G) = \frac{d\mathbb{G}_{1,n}(G)}{d\mathbb{G}_{0,n}} &= \frac{\frac{1}{|\tilde{\mathscr{L}}(v_n,n)|}\sum_{\phi\in\tilde{\mathscr{L}}(v_n,n)} \mathbb{G}_{1,n,\phi}(G)}{\frac{1}{|\tilde{\mathscr{L}}(v_n,n)|}\sum_{\phi\in\tilde{\mathscr{L}}(v_n,n)} \mathbb{G}_{0,n,\phi}(G)} \\ &= \frac{\frac{1}{|\tilde{\mathscr{L}}(v_n,n)|}\sum_{\phi\in\tilde{\mathscr{L}}(v_n,n)} \mathbf{1}_{\{\phi \text{ is a valid labeling for } H_1\}} q_0^{e(G)-e(H_1)}}{\frac{1}{|\tilde{\mathscr{L}}(v_n,n)|}\sum_{\phi\in\tilde{\mathscr{L}}(v_n,n)} \mathbf{1}_{\{\phi \text{ is a valid labeling for } H_0\}} q_0^{e(G)-e(H_0)}} \\ &= \frac{N(G,H_1)}{N(G,H_0)} \times q_0^{e(H_0)-e(H_1)}\end{aligned}$$

■

And from Lemma 5.1 in [JM15], we have

Lemma 8 *With the above setting, $\mathbb{G}_{0,n}$ and $\mathbb{G}_{1,n}$ are strongly distinguishable iff under $\mathbb{G}_{0,n}$*

$$\mathscr{L}(G) \xrightarrow{p} 0$$

They are not weakly distinguishable iff along some subsequences $\{n_k\}$

$$\mathscr{L}(G) \xrightarrow{p} 1$$

, while strongly distinguishable refers to two types of error converge to 0 and weakly distinguishable refers to the sum of two errors being less than 1.

Finally, before introducing our sufficient condition for distinguishing two subgraphs, we define a distortion for graphs as follow.

Definition 9 (graph distortion) *For two graphs G_0, G_1 with the same number of vertices, we define the edge distortion of G_0, G_1 as*

$$D_e(G_1,G_0) = e(G_0) - \max_{\phi\in\mathscr{L}(v_n,n)} |\{(i,j) : (i,j)\in G_0,\ (\phi(i),\phi(j))\in G_1\}|$$

And, define the distortion of G_0, G_1 as the maximum edge distortion of their subgraphs.

$$D(G_1,G_0) = \max_{S_0\subseteq G_0, S_1\subseteq G_1, |S_0|=|S_1|} D_e(S_1,S_0)$$

Note that the edge distortion and distortion of graphs are asymmetric. Here, we provide an easy example for you to get more intuition.

Example 10 (empty graph) *Consider G_0 an empty graph with n vertices and G_1 being arbitrary graph with n vertices. Then, $D_e(G_1,G_0) = e(G_1)$ and $D(G_1,G_0) = e(G_1)$.*

3.3 Sufficient condition for distinguishability

Recall that by Proposition 7 and Lemma 8, it suffices to show that under $\mathbb{G}_{0,n}$, $N(G,H_1) \xrightarrow{n\to\infty} 0$. The following Theorem 11

Theorem 11 (sufficient condition for distinguishability) *Consider $G \leftarrow \mathbb{G}_{0,n}$, we can distinguish $H_{0,n}$ and $H_{1,n}$ w.h.p as $n\to\infty$ if*

$$\frac{D(H_1,H_0)}{v_n} \geq \frac{\log n}{\log 1/q_0}$$

Proof: See Appendix C.3. ■

4 Conclusion

The hidden clique problem is indeed one of the most important problem in the study of average-case complexity. With a large number of applications in various disciplines, hidden clique problem can either be a hardness assumption or be an algorithm. For now, the goal is to close the computational gap.

As to the hidden subgraph problem, one can find out that a general characterization will be very difficult as long as we do not make any assumption on the structure of the subgraph. There are several candidate for categorizing the level of hardness, however, have not been well-studied and are nontrivial in the construction of lower bound. Moreover, the power of hidden subgraph problem is still unknown. Although it seems to be a generalization of hidden clique problem, the power might not increase. Also, if one want to apply it to real applications, he/she needs to identify useful subgraphs. As a result, this becomes a very application-specific issue.

4.1 Future works

In [ABW10], Applebaum et al. represented hidden clique problem in the form of DUE problem, which can be in some sense viewed as a hidden subgraph problem. The only difference is that DUE only requires certain structure of the graph instead of explicit hidden subgraph.

As the construction of lower bound is relatively difficult, in the future, I will first focus on the survey of various applications of hidden clique problem. By studying how people reduce hidden clique to their own usage, hopefully I can get more intuitions about the hardness of hidden subgraph problem. Also, maybe I can find some interesting applications and try to solve it with hidden clique/subgraph problems or prove its hardness via these conjectured hard problems.

References

[ABW10] Benny Applebaum, Boaz Barak, and Avi Wigderson. Public-key cryptography from different assumptions. In *Proceedings of the forty-second ACM symposium on Theory of computing*, pages 171–180. ACM, 2010.

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. *Random Structures and Algorithms*, 13(3-4):457–466, 1998.

[BR13] Quentin Berthet and Philippe Rigollet. Complexity theoretic lower bounds for sparse principal component detection. In *Conference on Learning Theory*, pages 1046–1066, 2013.

[CLR15] T Tony Cai, Tengyuan Liang, and Alexander Rakhlin. Computational and statistical boundaries for submatrix localization in a large noisy matrix. *arXiv preprint arXiv:1502.01988*, 2015.

[CX14] Yudong Chen and Jiaming Xu. Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices. *arXiv preprint arXiv:1402.1267*, 2014.

[DM15] Yash Deshpande and Andrea Montanari. Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems. *arXiv preprint arXiv:1502.06590*, 2015.

[FGR$^+$13] Vitaly Feldman, Elena Grigorescu, Lev Reyzin, Santosh Vempala, and Ying Xiao. Statistical algorithms and a lower bound for detecting planted cliques. In *Proceedings of the forty-fifth annual ACM symposium on Theory of computing*, pages 655–664. ACM, 2013.

[GV01] Dima Grigoriev and Nicolai Vorobjov. Complexity of null-and positivstellensatz proofs. *Annals of Pure and Applied Logic*, 113(1):153–160, 2001.

[HK11] Elad Hazan and Robert Krauthgamer. How hard is it to approximate the best nash equilibrium? *SIAM Journal on Computing*, 40(1):79–91, 2011.

[HWX14] Bruce Hajek, Yihong Wu, and Jiaming Xu. Computational lower bounds for community detection on random graphs. *arXiv preprint arXiv:1406.6625*, 2014.

[JM15] Hamid Javadi and Andrea Montanari. The hidden subgraph problem. *arXiv preprint arXiv:1511.05254*, 2015.

[Kar84] RM Karp. The probabilistic analysis of combinatorial optimization algorithms. In *Proceedings of the 10th International Symposium on Mathematical Programming*, 1984.

[Kuč95] Luděk Kučera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2):193–212, 1995.

[KZ14] Pascal Koiran and Anastasios Zouzias. Hidden cliques and the certification of the restricted isometry property. *Information Theory, IEEE Transactions on*, 60(8):4999–5006, 2014.

[Las01] Jean B Lasserre. Global optimization with polynomials and the problem of moments. *SIAM Journal on Optimization*, 11(3):796–817, 2001.

[MPW15] Raghu Meka, Aaron Potechin, and Avi Wigderson. Sum-of-squares lower bounds for planted clique. *arXiv preprint arXiv:1503.06447*, 2015.

[MW13] Raghu Meka and Avi Wigderson. Association schemes, non-commutative polynomial concentration, and sum-of-squares lower bounds for planted clique. In *Electronic Colloquium on Computational Complexity (ECCC)*, volume 20, page 105, 2013.

[MW$^+$15] Zongming Ma, Yihong Wu, et al. Computational barriers in minimax submatrix detection. *The Annals of Statistics*, 43(3):1089–1116, 2015.

[OZ13] Ryan O'Donnell and Yuan Zhou. Approximability and proof complexity. In *Proceedings of the Twenty-Fourth Annual ACM-SIAM Symposium on Discrete Algorithms*, pages 1537–1556. SIAM, 2013.

[Par00] Pablo A Parrilo. *Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization*. PhD thesis, Citeseer, 2000.

[PS$^+$00] Pavel A Pevzner, Sing-Hoi Sze, et al. Combinatorial approaches to finding subtle signals in dna sequences. In *ISMB*, volume 8, pages 269–278, 2000.

A Survey on hidden submatrix/clique problems

A.1 Introduction

This survey focus on the time, error trade-off problem of two problems: *hidden submatrix* and *hidden cluster*. Moreover, there also two kinds of different approaches to study these problems: *detection* and *localization*.

The goal here is to find the fundamental limit (w.r.t. size, sparsity) to achieve certain criteria. Traditional aspects consider the **feasible/infeasible** issues. That is, in what region can guarantee us to detect/localize the submatrix/cluster w.h.p. Furthermore, we want to find out in what region can we do so in a reasonable time or using a particle method to achieve the goals. Namely, where's the **tractable/intractable** region?

This week, I've surveyed the goals, motivations, and results of the following paper: [MW$^+$15], [CX14], [DM15], [CLR15], [HWX14], [BR13]. They focused on the time-error trade-off of hidden submatrix and cluster problem ([BR13] discussed the sparse PCA detection) with slightly different settings. I will show the difference and present their results.

A.1.1 Hidden submatrix problem (HSP)

There's a (symmetric) $n \times n$ matrix A where each of the entry $A_{i,j}$ follows the distribution (Gaussian, sub-Gaussian, etc.) with mean $\theta_{i,j}$. We say that A has a submatrix with size $k \times k$ (for simplicity, I consider the square submatrix case) if there is a index set $C, R \subset [n]$, $|C| = |R| = k$ such that

- If $i \in R,\ j \in C$, then $\theta_{i,j} \geq \lambda$.
- Else, $\theta_{i,j} = 0$.

, where λ can be thought of as the strength of signal.

Clearly that the detection problem is a hypothesis testing of whether there's a submatrix with size $k \times k$ and the localization version is to find the position of the submatrix.

Note that there are basically three parameters for us to tune with:

- Matrix size n
- Submatrix size (sparsity) k
- Signal strength λ

Intuitively, as $\frac{k}{n}$ becomes small, the problem becomes more difficult. And as λ grows large, the problem becomes easier.

HSP is discussed in [DM15], [CLR15], [MW$^+$15], [CX14].

A.1.2 Hidden clique/clustering problem (HCP)

Most of the HCP settings are based on the Erdos-Renyi random graph. Here, I introduce a general (multi-cluster) version of HCP[CX14]. Suppose there are n nodes and r cluster with k nodes in each cluster respectively. The connection rule is as follow:

- If u, v are in the same cluster, then they connect to each other with probability p.
- If u, v are **not** in the same cluster, then they connect to each other with probability q.

There are lots of variation by modifying the setting of p and q. For example:

- Planted r-Disjoint Clique: $p = 1,\ 0 < q < 1$
- Planted Densest Subgraph: $0 < q < p < 1$
- Planted Partition (Stochastic block model): $n = rk,\ p, q \in (0, 1)$
- Planted r-Coloring: $n = rk,\ 0 = p < q < 1$

The detection problem is to find out whether there is such a clique/cluster and the location problem is to find where it is.

There are basically four parameters for us to tune with:

- Number of nodes n
- Size of the clique k
- Number of cliques r
- Connection probability p, q

Intuitively, $\frac{k}{n}$ makes the problem harder as it becomes smaller, and the difference of p, q influences the difficulty of detection/localization.

HCP is an important model in communication network since it can describe the relationship among each node. It's discussed in [CX14], [DM15].

Remark: The best known polynomial time algorithm is $\Theta(\sqrt{n})$ and current state-of-the-art lower bound is $\Omega(\frac{n^{1/3}}{\log n})$ with SOS techniques in [DM15].

A.1.3 Statistical and computational concerns

Statistical point of view In the statistical context, we care whether the problem is feasible. That is, we want to know whether the problem can be solved w.h.p. under a given parameter setting. In other words, we want to find the region where there is a method for us to detect/localize the submatrix/clique with vanished error.

Note that here we not necessary consider the asymptotic scheme. Namely, some papers provide a condition for any finite parameter setting.

Computational point of view Here, we not only care about whether we can solve the problem or not, we care whether we can/cannot solve the problem **efficiently** (*i.e.,* polynomial time). Note that there are two slightly different philosophies:

- Achievable: provide a polynomial time algorithm to show the achievability.
- Non-achievable: use computational theoretic lower bound techniques to show the impossibility.

A.1.4 Common approaches

Here we don't focus on the statistical point of view since most of the results and techniques are well-studied. This section introduce the main methods used in proving the fundamental limit in a computational point of view.

Achievable [CX14] used convex relaxation MLE to achieve polynomial time localization for both HSP and HCP.

Non-achievable [BR13], [HWX14], [CLR15], and [MW$^+$15] used *hidden clique hypothesis* to show the impossibility of detect/localize sparse PCA, HSP, and HCP respectively.

On the other hand, [DM15] used *sum of square* (SOS) semidefinite hierarchy to improve the lower bound of HSP and HCP.

A.2 Recent results

In this section, I listed the results of the related papers without detail informations. The focus will be there methodologies, assumptions, and implications.

A.2.1 Improved sum-of-squares lower bounds for hidden clique and hidden submatrix problems [DM15]

This paper use SOS to provide a better lower bound for the detection HSP and HCP. They showed that SOS fails to detect

- **HCP**: $k \lesssim \frac{n^{1/3}}{\log n}$
- **HSP**: $k \lesssim \frac{\lambda^{-1} n^{1/3}}{\log n}$

Remark There's no poly-time algorithm known to detect HCP and HSP when $k = o(n^{1/2})$ but no one yet to prove it. The above lower bound is the state of the art.

We will discuss more on SOS later.

A.2.2 Computational and statistical boundaries for submatrix localization in a large noisy matrix [CLR15]

This paper focus on the submatrix localization problem in a noisy setting. As a result, here we consider the signal to noise ratio SNR $:= \frac{\lambda}{\sigma}$. And they provided the following results in Figure 3 The two thresholds are:

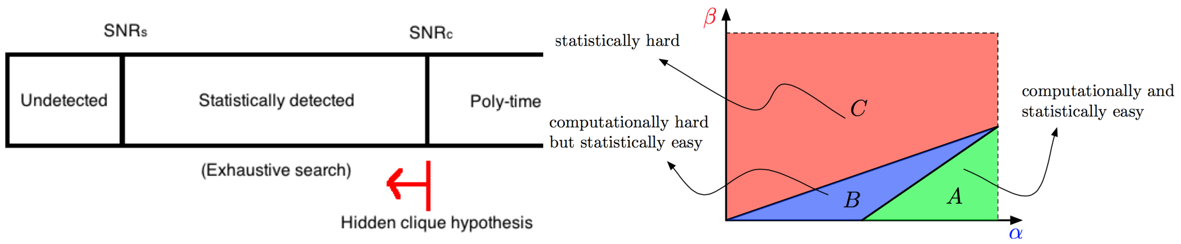

Figure 3: [CLR15]

- $\text{SNR}_s \asymp \sqrt{\frac{\log n}{k}}$

- $\text{SNR}_c \asymp \sqrt{\frac{n}{k^2}} + \sqrt{\frac{\log n}{k}}$

We can parametrize the model with n: $k = \Theta(n^{\alpha})$, $\frac{\lambda}{\sigma} = \Theta(n^{-\beta})$

A.2.3 Computational barriers in minimax submatrix detection [MW+15]

This paper discusses the submatrix detection problem. They parametrized with n: $k = \Theta(n^{\alpha})$, $\lambda = \Theta(n^{-\beta})$ and got the following results:

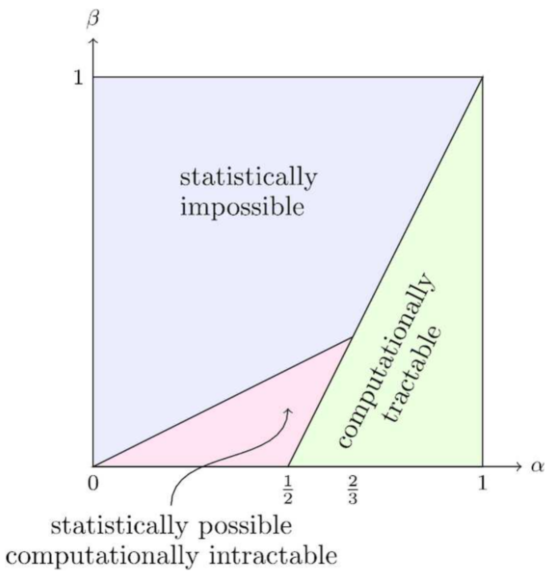
β
1
statistically impossible
computationally tractable
0
$\frac{1}{2}$
$\frac{2}{3}$
1
α
statistically possible computationally intractable

Figure 4: [MW+15]

They used discretized Gaussian model to approximate the continuous setting. Also, the computational lower bound is provided by hidden clique hypothesis. Also, the authors pointed out that the results will be different when using different loss function.

Remark There are known necessary and sufficient condition for the possibility of detection (feasibility):

- Error $\to 0$ if $\frac{\lambda}{n/k^2} \to \infty$
- Error $\to 1$ if $\frac{\lambda}{n/k^2} \to 0$

A.2.4 Statistical-computational tradeoffs in planted problems and submatrix localization with a growing number of clusters and submatrices [CX14]

This paper defined four regions:

- **Impossible region**: Infeasible any algorithm.
- **Hard region**: Feasible for statistical algorithm.
- **Easy region**: Feasible for polynomial time algorithm.
- **Simple region**: Feasible for counting/threshold algorithm.

The following consider HCP and HSP respectively.

HCP They parametrized with p and q: $p = 2q = \Theta(n^{-\alpha})$, $k = \Theta(n^{\beta})$. And the quantity that captures the hardness is SNR $:= \frac{(p-q)^2}{q(1-q)}$.

HSP They parametrized with n: $\lambda^2 = \Theta(n^{-\alpha})$, $k = \Theta(n^{\beta})$. And the quantity that captures the hardness is SNR $:= \lambda^2$.

The four regions for both problems are characterized with SNR in Figure 5.

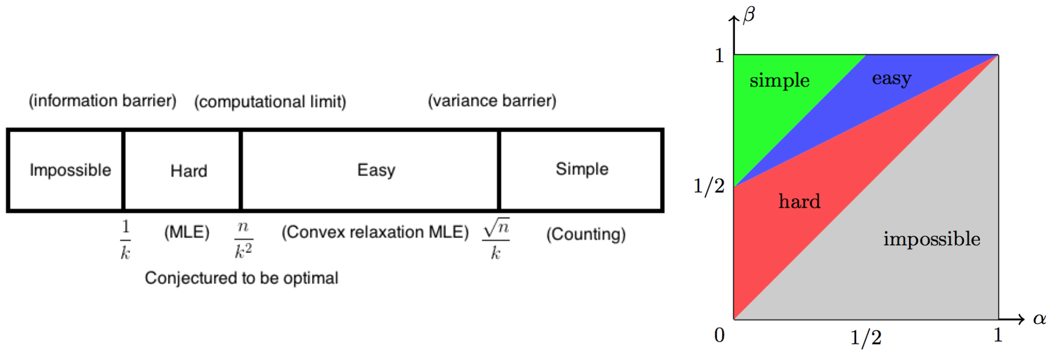

Figure 5: [CX14]

B Sum of square method

B.1 SOS

SOS method aims to answer the question:

When do equations and inequalities have real solutions?

SOS method answers the above question with two different directions: *pseudo distribution* and *Positivstellensatz*. Generally speaking, pseudo distribution gives a **close answer** that satisfies certain convex constraint and Positivstellensatz provides a **infeasible certificates** for real solutions to system of polynomial equations. The infeasible certificate is always correct and the pseudo distribution will have higher probability to be correct as long as the degree of SOS algorithm becomes higher.

In this section, we start with the motivation of SOS method, then provide a high-level overview. Next, we will focus on how to use SOS to conduct a lower bound proof. And in the next section, we will take planted clique problem as an example.

B.1.1 Motivation: Semidefinite Programming (LMIs)

Semidefinite programming (SDP), a.k.a. linear matrix inequality (LMI), is a general convex optimization problem:

$$\begin{array}{ll} \min & \mathbf{tr}(CX) \\ \text{subject to} & \mathbf{tr}(A_iX) = b_i,\ i = 1,...,p \\ & X \geq 0 \end{array}$$

The importance of SDP is that it can be solved in **polynomial time**! (via interior point method) Moreover, with some modification, SDP can be written in an equivalent form as follow:

$$\begin{array}{ll} \min & c^T x \\ \text{subject to} & A_0 + x_1A_1 + ... + x_nA_n \geq 0 \end{array}$$

And the constraint is equivalent to

$$\exists x \forall y P(x,y) \geq 0$$

, where $P(x,y) = y^T(A_0 + x_1A_1 + ... + x_nA_n)y$. We can see that this is actually a Σ_2 problem! Intuitively, we can somehow solve a class of Σ_2 problem in polynomial time. As a result, we want to characterize a general purpose method to keep the tractability to these problem.

Intuition (LMI)

Linear matrix inequalities (LMIs) are

quadratic forms that are nonnegative

Note that, although the problem is in Σ_2, they still can be solved in polynomial time.

From LMIs, we can generalize the idea to SOS. Formally, we define a multivariate **polynomial** $p(x)$ to be SOS if

$$p(x) = \sum_i q_i^2(x)$$

, and clearly that if a polynomial $p(x)$ is SOS, then $p(x) \geq 0$, $\forall x$. Thus, we define a SOS program to be:

$$\begin{array}{ll} \text{min} & c_1u_1 + ... + c_nu_n \\ \text{subject to} & P_i(x,u) := A_{i0}(x) + A_{i_1}(x)u_1 + ... + A_{in}(x)u_n \text{ is SOS} \end{array}$$

Intuition (SOS)

SOS is a

affine families of polynomials that are sum of squares

Moreover, we can solve SOS program in polynomial time.

Note that just using **nonnegative polynomials** as constraints is a NP-hard problem and here as we turn to SOS program, the complexity is in P! Moreover, in several important cases (quadratic, univariate,...), nonnegativity and SOS are the same.

B.1.2 Introduction

To illustrate the main idea of SOS method, let's start with a imaginary game. Suppose there is a difficult problem $\mathscr{P}$ and there are two people want to find out whether $\mathscr{P}$ is correct or not. The optimist want to show the correctness of $\mathscr{P}$. So what he can do is to find an evidence to support his belief. On the other hands, the pessimist wants to refute the problem. His goal is to find a infeasible certificate to show that $\mathscr{P}$ is wrong.

In this traditional setting $\mathscr{P}$ is like all other decision problem. However, the main idea of SOS method here is to relax the requirement for the optimist. Namely, it's acceptable if the optimist can only provide a **close evidence to certain degree** r: *pseudo distribution*. Intuitively, this relaxation trade-off the soundness of the algorithm with time, since the algorithm will run in $O(n^r)$.

On the other hands, the pessimist has a general strategy too: *Positivstellensatz*. It is a very strong semidefinite program that generates a degree-r SOS infeasible proof as long as long as there's such proof! Note that both pseudo distribution and Positivstellensatz are convex optimization program (semidefinite programming), which is solvable in polynomial time. And the key idea here is that they are actually **dual** to each other![Las01],[Par00] The following table summarize this imaginary game:

Optimist	Close evidence	*Pseudo distribution*
Pessimist	Valid refutation	*Positivstellensatz*

Table 2: Imaginary game of SOS method.

There are two usage of SOS method: SOS algorithm and SOS lower bound proof. The former care about how to use a pseudo distribution to approximate a real solution. The latter focus on showing that there is no Positivstellensatz proof on certain degree, or equivalently showing that there's a pseudo distribution. Basically, the SOS algorithm and SOS lower bound proof have the following concepts:

1. For a problem, we can form a set of polynomial **axioms**.

$$\mathscr{E} = f_1(x) = 0,\ f_2(x) = 0,\ ...,\ f_m(x) = 0$$

   and then we can use the following *rules of inference* to derive some inequalities

   - $p \geq 0, q \geq 0 \vdash p + q \geq 0$
   - $p \geq 0, q \geq 0 \vdash pq \geq 0$
   - $p \geq 0 \vdash p^2 \geq 0$

   Remark: We can always replace inequality $P(x) \geq 0$ by equation $P(x) - y^2 = 0$ for some slack variable y.

2. If we can use these axioms to derive some contradiction through the above rules of inference, then we can certify infeasibility of the problem. Most of the time, we

derive the contradiction in the following form:

$$\text{positivstellensatz refutation } (\mathbf{PS}(\text{r})): \sum_{i=1}^{m} f_i g_i = 1 + \sum_{i=1}^{N} h_i^2$$

, where $g_1,...,g_m$ and $h_1,...,h_N$ are some arbitrary polynomials and $deg(f_i g_i) \leq 2r$.

In other words, the above process shows the **infeasibility** of a system or the failure of a problem. As a result, it turns out to be a general algorithm to decline a problem. We call this a degree$-r$ SOS proof.

However, what's the importance of such idea?

Actually, it is because that SOS hierarchy has some connection with the above proof concept and can be stated as the following theorem:

Theorem 12 (SOS) *Under some mild conditions, there is a $n^{O(r)}$ time algorithm that given a set of polynomial axioms $\mathscr{E}$ and either output*

- *A degree$-r$ pseudo-distribution μ consistent with $\mathscr{E}$*

or

- *A degree$-r$ SOS proof that $\mathscr{E}$ is unsatisfiable.*

Intuition (degree)

When $r = 1$, this is actually a *linear programming*. When $r = 2$, this is a *semi-definite programming*. And when $r = n$, this is a *brute force/exhaustive search* algorithm!
As a result, what we are interested in is r in the range $2 < r \ll n$.

Here, we don't explain the concept of **pseudo-distribution** too deep. Intuitively, it is an object that closely related to the problem instance and obey the axioms $\mathscr{E}$. However, we cannot use a pseudo-distribution to certify the correctness of the problem.

Example: Now, let's take MAX-CLIQUE as an example:
Let $G = (V,E)$ be a graph and $V = [n]$. Let $x_i = \mathbf{1}_{\{i \in k-clique\}}$, we can construct the

following axioms:

$$\begin{array}{lll}(\textsc{Max-Clique}): & x_i^2 - x_i,\ \forall i \in V & x_i = 0,1 \\ & x_i x_j,\ \forall (i,j) \notin E & \text{if } (i,j) \notin E\text{, either } i \text{ in } k\text{-clique or } j \\ & k - \sum_{i \in V} x_i & \text{there are } k \text{ vertices in the } k\text{-clique}\end{array}$$

[MW13] showed that with high probability there's no such infeasible proof for $k \leq \frac{\sqrt{n}}{(C\log n)^{r^2}}$. And [DM15] sharpened the result to $k \leq \frac{Cn^{1/3}}{\log n}$ for degree-4 SOS relaxation. In other words, Deshpande and Montanari showed that there's no $O(n^4)$ SOS algorithm to solve Max-Clique as $k \leq \frac{n^{1/3}}{\log n}$.

Intuition (Positivstellensatz refutation and SOS lower bound)

- SOS algorithm:
  - Positivstellensatz refutation provides a general way to decline a problem.
  - SOS can certify the infeasibility in $O(n^{\Theta(r)})$ time.
- SOS lower bound:
  - We can show that there's **no PS**(r) refutation and thus there's no r round SOS algorithm to show the indistinguishability and thus provide a lower bound.
  - The lower bound is in the sense that SOS algorithm fails to solve the problem in some small degree.

Remark: SOS algorithm and SOS lower bound are working in the different directions but share the same core idea: Positivstellensatz refutation.

B.1.3 Proof structure

Simply speaking, here we want to present a lower bound for **PS**(r). And first we need to define the polynomials set that we work on. Let $\mathscr{P}(n,2r) : \{f : \mathbb{R}^n \rightarrow \mathbb{R}\}$ being a set of n-variate polynomials with degree at most $2r$. Next, we would like to have a definition for mapping that captures the non-negativity of square polynomial. Thus, we define PSD mapping as

Definition 13 (PSD mapping) *A linear mapping $\mathscr{M} : \mathscr{P}(n,2r) \rightarrow \mathbb{R}$ is a PSD mapping if $\mathscr{M}(P^2) \geq 0$ for all n-variate polynomial with degree at most r.*

Moreover, we need a mapping to quickly check whether a polynomial is nonnegative. Thus, we call a PSD mapping that preserves the zeroness as dual certificate

Definition 14 (dual certificate) *Given a set of axioms $\{f_1, f_2, ..., f_m\}$, a dual certificate for these axioms is a PSD mapping $\mathscr{M}$ such that $\mathscr{M}(f_i g) = 0$ for all f_i in the axiom and polynomial g where $deg(f_i g) \leq 2r$.*

Remark: Actually, the pseudo-distribution mentioned in Theorem 12 is just a general name for dual certificate!

And there's a lemma that connects the existence of dual certificate and $\mathbf{PS}(r)$ refutation:

Lemma 15 *Given a set of axioms $\{f_1, f_2, ..., f_m\}$, there does not exist a $\mathbf{PS}(r)$ refutation if there exists a $(n, 2r)$-dual certificate.*

The reason why we can find a dual-certificate as long as the $\mathbf{PS}(r)$ refutation exists is because of the **dual property** semidefinite programming. Intuitively, finding a dual certificate is equivalent to show a lower bound for $\mathbf{PS}(r)$ refutation. That is, as long as we can find a dual certificate, we can never turn down a infeasible problem!

Conclusion

To sum up, the proof flow can be simplified as follow:

1. Write down axioms of the given problem $\rightarrow$ Polynomial constraints $\{f_i\}$
2. Turn into constraints for dual certificate $\rightarrow$ Guess a natural solution to the dual certificate $\mathscr{M}$.
3. Show that $\mathscr{M}$ is PSD w.h.p $\rightarrow$ Show the PSDness of M is sufficient.
   (a) $E[M]$ is PSD
   (b) Reduction to PSDness of M'_r, the principle minor of M and bound the mean and variance of M'_r.

B.1.4 Pseudo-distribution

Pseudo-distribution is a crucial concept in SOS proof. It can be regarded as a **computationally -bounded** observer. Concretely, the generation of a pseudo-distribution does not consume every possible computational resource. Moreover, the the results must satisfy problem constraints and SOS nonnegativity. Formally speaking, the expectation of pseudo distribution on the axiom polynomials must be 0 and the expectation of all square polynomial must be nonnegative.

The algorithm only search on a low degree of possibility. Somehow, this captures our limited computational abilities. For convenience, we can simply think of a pseudo-distribution as our ”computational knowledge”.

B.1.5 Resources

1. **Lectures:**
   - Barak SOS lecture: link
   - Laurent Semidefinite optimization lecture: link
   - Parrilo Algebraic Techniques and Semidefinite Optimization lecture: link
2. **Videos**:
   - TCS+ talk - Boaz Barak: link
   - TCS+ talk - Aaron Potechin: link
   - ICM2014 - Boaz Barak: link
3. **Papers**:
   - Complexity of Null-and Positivstellensatz proofs: [GV01]
   - Global optimization with polynomials and the problem of moments: [Las01]
   - Structured semidefinite programs and semialgebraic geometry methods in robustness and optimization: [Par00]
   - Association schemes, non-commutative polynomial concentration, and sum-of-squares lower bounds for planted clique: [MW13]
   - Approximability and proof complexity: [OZ13]

B.2 Example: SOS Lower Bound For Planted Clique Problem

In this section, we see a SOS lower bound example in planted clique problem. The proof is provided by [MW13]. Meka and Wigderson constructed a special moment matrix as a dual certificate to show the impossibility of refutation in some low degree.

B.2.1 Problem settings & Goal

First, let's recall the problem setting:

$$\begin{array}{lll} (\textsc{Max-Clique}): & x_i^2 - x_i,\ \forall i \in V & x_i = 0,1 \\ & x_i x_j,\ \forall (i,j) \notin E & \text{if } (i,j) \notin E \text{, either } i \text{ in } k\text{-clique or } j \\ & k - \sum_{i \in V} x_i & \text{there are } k \text{ vertices in the } k\text{-clique} \end{array}$$

1. $M_{I,J} = \mathscr{M}(\prod_{s \in I \cap J} x_s)$, where I,J are subsets with size no larger than r.

$$M \text{ is PSD} \Leftrightarrow \mathscr{M} \text{ is PSD}$$

   Here, M is the association scheme for $\mathscr{M}$. We can utilize some good properties and results of it from combinatorial theory, which will be introduced in latter section.

2. $X_I = \prod_{s \in I} x_s$, where $I \subseteq [n]$.

We want to show the following two results: ([MW13])

Theorem 16 (planted clique lower bound)

1. *(Maximum clique) W.h.p., for $G \leftarrow G(n,1/2)$ the natural r-round SOS relaxation of the maximum clique problem has an integrallity gap of at least $\sqrt{n}/(C\log n)^{r^2}$.*
2. *(Planted clique) W.h.p., for $G \leftarrow G(n,1/2,t)$ the natural r-round SOS relaxation of the maximum clique problem has an integrallity gap of at least $\sqrt{n}/t(C\log n)^{r^2}$.*

B.2.2 Meka-Wigderson's candidate for dual certificate

With the above axioms, for any suitable dual certificate $\mathscr{M}$ for graph G should satisfies

1. $\mathscr{M}(X_I) = 0$, $\forall I,\ |I| \leq 2r$, I is not a clique in G.
2. $\mathscr{M}((\sum_i x_i - k)X_I) = 0$, $\forall I,\ |I| \leq 2r$.

With these two constraints, Meka and Wigderson then guess the following candidate for dual certificate:

Proposition 17 *Define a candidate of dual certificate for graph G as $\mathscr{M}$. For all $I \subseteq [n]$ and $|I| \leq 2r$, let*

$$\mathscr{M}(\prod_{i \in I} x_i) = deg_G(I) \cdot \frac{\binom{k}{|I|}}{\binom{2r}{|I|}}$$

Remark:

1. $deg_G(I)$ = the number of size $2r$ clique in G that contains I. Or equivalently, $deg_G(I) = |\{S \subseteq [n] : I \subseteq S, |S| = 2r, S \text{ is a cliquie in } G\}|$.
2. Since $\mathscr{M}$ is a mapping from polynomial of degree at most $2r$ to a real number, it's sufficient to specify all the possible monomial with degree no more than $2r$.
3. We can check that this $\mathscr{M}$ satisfies the above constraints. (page 16 in [MW13])

Now, to prove Theorem 16, it's sufficient to show the PSD'ness of $\mathscr{M}$. And from the previous section, we also know that it's equivalent to show the following matrix is PSD:

$$M(I,J) = deg_G(I \cup J) \cdot \frac{\binom{k}{|I \cup J|}}{\binom{2r}{|I \cup J|}}$$

Intuition (Meka-Wigderson's moment matrix)

Consider the moment of pseudo distribution:

$$\tilde{\mathbb{E}}(X_I) = \mathscr{M}(I)/deg_G(\emptyset) = \frac{\binom{k}{|I|}}{\binom{2r}{|I|}} \times \frac{deg_G(I)}{deg_G(\emptyset)}$$

Observe that if I is a **clique**

- $deg_G(I) \sim n^{d-|I|}$
- $\binom{k}{|I|} \sim k^{|I|}$
- $deg_G(\emptyset) \sim n^d$
- $\binom{2r}{|I|} \sim constant$

Thus,

$$\tilde{\mathbb{E}}(X_I) \sim \frac{k^{|I|}}{n^{|I|}}$$

The Meka-Wigderson moment matrix will look like follow:

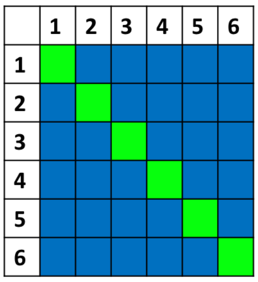

Figure 6: $r = 1$

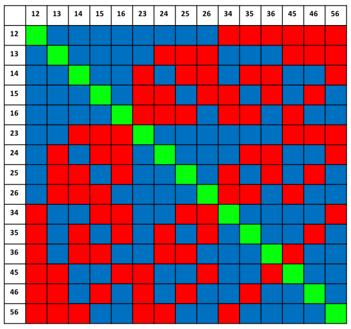

Figure 7: $r = 2$

Figure 1:

- 🟩: $M(i,i) \approx \frac{k}{n}$
- 🟦: If there's an edge between i and j, then $M(i,j) \approx \frac{k^2}{n^2}$.Else, $M(i,j) = 0$.

Figure 2:

- 🟩: If there's an edge between i and j, then $M(\{i,j\},\{i,j\}) \approx \frac{k^2}{n^2}$. Else, $M(\{i,j\},\{i,j\}) = 0$.
- 🟦: If there's a clique among i,j and k, then $M(\{i,j\},\{i,k\}) \approx \frac{k^3}{n^3}$. Else, $M(\{i,j\},\{i,k\}) = 0$.
- 🟥: If there's a clique among i,j,k and l, then $M(\{i,j\},\{k,l\}) \approx \frac{k^4}{n^4}$. Else, $M(\{i,j\},\{k,l\}) = 0$.

B.2.3 Proof flow

To prove the lower bound of maximum clique and planted clique problems through SOS lower bound techniques, we simply need to do two things:

1. Design a dual certificate $\mathscr{M}$ based on clique axioms.
2. Show that $\mathscr{M}$ is PSD.

With these two statements, we can thus claim that the problem does not have a degree $2r$ SOS algorithm.

However, the second step is not easy, most of the technical parts lie in there. In the following example of [MPW15], they prove the PSD'ness of $\mathscr{M}$ with the following substeps:

1. Relate $\mathscr{M}$ to a matrix representation M such that $\mathscr{M}$ is PSD $\Leftrightarrow$ M is PSD.
2. Add small value to the nonzero entries of M and get M'.
    - If M' is PSD, then M is PSD.
    - Write $M' = E + L + \Delta$, where $E = \mathbb{E}[M']$, L is a special local random matrix and Δ is a small global random matrix.
3. Show M' is PSD:
    - $E \succ 0$ and $\lambda_{\min}(E) = \Omega(k^r n^r)$
    - $||L|| < Ck^{2r}n^{r-0.5}\log n$
    - $||\Delta|| < Ck^{2r}n^{r-0.5}\log n$

B.2.4 Johnson scheme

Johnson scheme is a special matrix whose columns and rows refer to a subset of set S. More interestingly, the value of the entry say $M(I,J)$ only depends on the size of $|I \cup J|$. This beautiful structure immediately leads to some good properties especially in the upper bound of eigenvalue. As we want to show the PSDness of matrices, it's sufficient for us to bound their least eigenvalue. And we are going to utilize Johnson scheme to help us do so.

To define Johnson scheme, we first need to introduce one concept: *set symmetry*.

Definition 18 (set symmetry) *We say a matrix $M \in \mathbb{R}^{\binom{[n]}{r} \times \binom{[n]}{r}}$ is set symmetric if for every subset $I,J \in \binom{[n]}{r}$, $M(I,J)$ only depends on $|I \cap J|$.*

Intuitively, the matrix depends only on the size of intersection.

Definition 19 (Johnson scheme) *For n and $r \leq n/2$, let $\mathscr{J} \equiv \mathscr{J}_{n,r} \subseteq \mathbb{R}^{\binom{[n]}{r} \times \binom{[n]}{r}}$ be the subspace of all set symmetry matrix. $\mathscr{J}$ is called the Johnson scheme.*

Clearly, we can see that the rank of Johnson scheme is $r+1$ because the size of intersection can only be ranged from 0 r. Now, we might wonder is there a good basis for Johnson scheme. The following shows two common bases:

- (**D-Basis**) For $0 \le l \le r$,
$$D_l(I,J) = \mathbf{1}_{|I\cap J|=l}$$
That is, the indicator of the size of intersection.
- (**P-Basis**) For $0 \le t \le r$.
$$P_t(I,J) = \binom{|I\cap J|}{t}$$
That is, the number of possible subset of the intersection of I and J.

And the transformation of D-Basis and P-Basis are:

- $P_t = \sum_l \binom{l}{t} D_l$
- $D_l = \sum_t (-1)^{t-l} \binom{t}{l} P_t$

Now, with these two bases, we want to further characterize the eigenvalues of matrices in Johnson scheme. And here we first present some well-known facts:

Lemma 20 *The matrices in Johnson scheme commute to each other. In other words, they are* **simultaneously diagonalizable**.

Lemma 21 *There are subspaces $V_0,...,V_r \in \mathbb{R}^{\binom{[n]}{r}\times\binom{[n]}{r}}$ such that*

1. *$V_0,...,V_r$ are eigenspaces of $\mathscr{J}$ and orthogonal to each other.*
2. *$dim(V_j) = \binom{n}{j} - \binom{n}{j-1}$. Thus, $\sum_j dim(V_j) = \binom{n}{r}$.*
3. *The eigenvalues of P_t corresponds to eigenspace V_j is*
$$\lambda_j(P_t) = \mathbf{1}_{j\le t}\binom{n-t-j}{r-t}\cdot\binom{r-j}{t-j}$$
4. *For arbitrary matrix $Q \in \mathbb{R}^{\binom{[n]}{r}\times\binom{[n]}{r}}$, $Q = \sum_l \alpha_l D_l$, and $\beta = \sum_{l\le t}\binom{l}{t}\alpha_l$, where $\alpha_l \ge 0$. Then,*
$$\lambda_j(Q) \le \sum_{t\ge j} \beta_t \cdot \binom{n-t-j}{r-t}\cdot\binom{r-j}{t-j}$$

C Proofs of current works

C.1 Proof of Theorem 3

Here, we only prove the lower bound of node-flipping error since the other proofs can be found in Appendix C.2 in a more general version.
First, by Janson's inequality, we have

$$\begin{aligned}
P[(\mathscr{E}^{node})^C] &= P[\{R \neq \emptyset\}^C] = P[\cap_i \{R_i \neq \emptyset\}^C] = P[\cap_i (\mathscr{E}_i)^C] \\
(\because \text{Janson's inequality}) &\leq \prod_i P[(\mathscr{E}_i)^C] \cdot e^{\Delta/(1-q_0)^2} \\
&= \prod_i [1 - q_0^{d_{H_n}(i)}]^{n-|H_n|} \cdot e^{\Delta/(1-q_0)^2} \\
&\leq \prod_i [e^{-q_0^{d_{H_n}(i)}}]^{n-|H_n|} \cdot e^{\Delta/(1-q_0)^2} \\
&\leq e^{(n-|H_n|)\cdot \sum_i -q_0^{d_{H_n}(i)} + \Delta/(1-q_0)^2} \\
&\leq e^{-(n-|H_n|)\cdot |H_n| \cdot q_0^{d_{max}} + \Delta/(1-q_0)^2}
\end{aligned}$$

, where, by Janson's inequality, $\Delta = \sum_i \sum_{i' \sim i} P[(\mathscr{E}_i)^C \cup (\mathscr{E}_{i'})^C]$. Note that since each node in H_n cannot have more than d_{max} neighbors, we have a direct loose upper bound on Δ.

$$\Delta \leq \frac{|H_n|}{2} d_{max}^2$$

Thus,

$$\begin{aligned}
P[R = \emptyset] &\leq e^{-(n-|H_n|)\cdot|H_n|\cdot q_0^{d_{max}} + \frac{|H_n|\cdot d_{max}^2}{2(1-q_0)^2}} \\
&= e^{|H_n|\cdot[\frac{d_{max}^2}{2(1-q_0)^2} - (n-|H_n|)\cdot q_0^{d_{max}}]}
\end{aligned}$$

C.2 Proof of Theorem 5

- For single-to-single set-flipping error

$$\begin{aligned}
P[R_{SS'} \neq \emptyset] &\leq q_0^{|\partial S| + |E(S)|} |S|! = q_0^{|S|\cdot(\frac{|\partial S|}{|S|} + \frac{|E(S)|}{|S|})} |S|! \\
&\leq q_0^{\alpha |S|} |S|!
\end{aligned}$$

- For single set-flipping error

$$\begin{aligned} P[R_S \neq \emptyset] &\leq P[\cup_{S' \subseteq V(G_n)\setminus V(H_n):|S'|=|S|} R_{SS'} \neq \emptyset] \\ (\because \text{union bound}) &\leq \sum_{S' \subseteq V(G_n)\setminus V(H_n):|S'|=|S|} R_{SS'} P[R_{SS'} \neq \emptyset] \\ &\leq \binom{n-|H_n|}{|S|} q_0^{\alpha|S|}|S|! \leq n^{|S|} q_0^{\alpha|S|} \end{aligned}$$

- For set-size set-flipping error

$$\begin{aligned} P[R_k \neq \emptyset] &= P[\cup_{S \subset V(H_n):|S|=k} R_S] \\ (\because \text{union bound}) &\leq \binom{|H_n|}{k} n^k q_0^{\alpha k} \leq \frac{n^k |H_n|^k q_0^{\alpha k}}{k!} \\ &= \frac{(n \cdot |H_n| \cdot q_0^{\alpha})^k}{k!} \end{aligned}$$

- For set-flipping error

$$\begin{aligned} P[R^e] &= P[\cup_{k=1}^{|H_n|} R_k \neq \emptyset] \\ (\because \text{union bound}) &\leq \sum_{k=1}^{|H_n|} P[R_k \neq \emptyset] \leq \sum_{k=1}^{|H_n|} \frac{(n \cdot |H_n| \cdot q_0^{\alpha})^k}{k!} \\ &\leq \sum_{k=0}^{|H_n|} \frac{(n \cdot |H_n| \cdot q_0^{\alpha})^k}{k!} - 1 \\ &\leq \sum_{k=0}^{\infty} \frac{(n \cdot |H_n| \cdot q_0^{\alpha})^k}{(k!)} - 1 = e^{n \cdot |H_n| \cdot q_0^{\alpha}} - 1 \end{aligned}$$

C.3 Proof of Theorem 11

First, observe that for any labeling $\phi \in \tilde{\mathbb{L}}(v_n, n)$,

$$\mathbb{G}_{0,n}(\phi \text{ is a valid labeling for } H_1) \leq q_0^{D(H_1,H_0)}$$

Then, using union bound,

$$\begin{aligned} \mathbb{G}_{0,n}[\sum_{\phi \in \tilde{\mathscr{L}}(v_n,n)} \mathbf{1}_{\{\phi \text{ is a valid labeling for } H_1\}} \neq 0] &\leq |\tilde{\mathscr{L}}(v_n,n)| \times q_0^{D(H_1,H_0)} \leq n^{v_n} \times q_0^{D(H_1,H_0)} \\ &= e^{v_n \log n - D(H_1,H_0)\log 1/q_0} \\ &= e^{v_n \log n [1 - \frac{D(H_1,H_0)\log 1/q_0}{v_n \log n}]} \end{aligned}$$

As $\frac{D(H_1,H_0)\log 1/q_0}{v_n \log n} > 1$, $\mathbb{G}_{0,n}[\sum_{\phi\in\tilde{\mathscr{L}}(v_n,n)} \mathbf{1}_{\{\phi \text{ is a valid labeling for } H_1\}} \neq 0] \xrightarrow{n\to\infty} 0$. Thus, $D(H_1,H_0) > \frac{\log n}{\log 1/q_0}$ is a sufficient condition for distinguishability.