ΕΘΝΙΚΟ ΚΑΠΟΔΙΣΤΡΙΑΚΟ ΠΑΝΕΠΙΣΤΗΜΙΟ
ΤΜΗΜΑ ΜΑΘΗΜΑΤΙΚΩΝ

ΜΕΤΑΠΤΥΧΙΑΚΟ ΠΡΟΓΡΑΜΜΑ ΛΟΓΙΚΗΣ ΚΑΙ ΑΛΓΟΡΙΘΜΩΝ

Διπλωματική Εργασία

Εύρεση Κλικών σε Τυχαίους Γράφους

του

Ανδρέα Γαλάνη

Επιβλέπων Καθηγητής:
Ευστάθιος Ζάχος

17 Δεκεμβρίου 2010

Abstract

The problem of finding cliques in arbitrary graphs is a well known NP-complete problem. From work culminating in the powerful theory of PCPs, it is now known that solving the clique problem even within ratio $n^{1-\epsilon}$ is NP-hard. However, we can recover a maximum clique if the input graph is drawn randomly from a distribution of graphs with n vertices which contain a sufficiently large planted solution. Specifically, we restrict ourselves to one of the most well studied distributions where the input graph is generated as follows: a random set of k vertices is first selected and forced into a clique. Then, any other edge is included independently with probability $1/2$. The goal is to recover the maximum clique. In this Master's thesis, we present the current state-of-the art for the planted clique problem.

Ευχαριστίες

Αυτή η διπλωματική δεν θα μπορούσε να είχε πραγματοποιηθεί χωρίς τη συμβολή του κ. Ζάχου σε πολλά επίπεδα. Ως καθηγητής στα προπτυχιακά μαθήματα της σχολής, μου έδωσε τα ερεθίσματα να ασχοληθώ με την επιστήμη των υπολογιστών. Ως επιβλέπων της διπλωματικής αυτής, με ενθάρρυνε και μου έδωσε τη δυνατότητα να ασχοληθώ με ένα πολύ ενδιαφέρον και πολύπλευρο θέμα. Και τέλος, αλλά ίσως και πιο σημαντικά, με έχει επηρεάσει ουσιαστικά και με τον καλύτερο δυνατό τρόπο.

Θα ήθελα επίσης να ευχαριστήσω τα άλλα δύο μέλη της τριμελούς εξεταστικής επιτροπής, κ. Παγουρτζή και κ. Φωτάκη, που εκτός από διδάσκοντές μου, με έχουν καθοδηγήσει καθόλη τη διάρκεια της προσπάθειας μου.

Ευχαριστώ επίσης την εργαστηριακή ομάδα του Corelab για το ευχάριστο και δημιουργικό κλίμα που πάντα προσφέρει.

Τέλος, θέλω να ευχαριστήσω τους γονείς μου, που σε κάθε φάση της ζωής μου με στηρίζουν με τον καλύτερο δυνατό τρόπο και εγώ δεν τους δείχνω ποτέ τι σημαίνουν για μένα.

Contents

Chapter 1

The case of Random graphs

1.1 Introduction

The clique problem in graphs asks for the maximum subset of vertices which are mutually adjacent. It is one of the most well known NP-complete problems ([Kar72]). By now, much more is known about the hardness of clique. A series of results, starting with the seminal work of Hastad in query efficient PCP verifiers([Hås97]), has finally lead to a proof that approximating the clique problem even within ratio $n^{1-\epsilon}$ for any $\epsilon > 0$ is NP-hard ([Zuc06]).

On the other hand, in an attempt to capture real life instances that emerge in practice (e.g. networks), Erdös and Rényi introduced the $\mathcal{G}(n, p)$ distribution on graphs. While the $\mathcal{G}(n, p)$ model as a "real life instance" has certain limitations, it is still cosidered a great basis to study graph properties.

This Master's thesis focuses on the restriction of the clique problem to the distribution $\mathcal{G}(n, p)$ and displays algorithmic ideas which perform well on average case instances.

We have tried to make the presentation as self-contained as possible. Still, a basic background in linear algebra and probabilistic arguments is assumed.

1.2 Cliques in Random Graphs

Consider the following process to generate a graph G on n vertices: include each of the $\binom{n}{2}$ edges independently with probability $1/2$. Denote the relulting distribution as $\mathcal{G}(n, 1/2)$. Given $G \sim \mathcal{G}(n, 1/2)$, we are going to address the size of the maximum clique in G. This is captured by the

following theorem.

Theorem 1.1. *In almost every graph $G \sim \mathcal{G}(n, 1/2)$ the largest clique has size $(2 + o(1)) \log n$.*

While Theorem 1.1 is well known and can be proved using standard means, we include a proof which addresses some of its details which are usually omitted. In addition, the proof will yield that the maximum clique number can take a finite number of values, which was first proved in a stronger sense by [BE76].

Proof (of Theorem 1.1) Let X_k be the number of k-cliques in a graph G drawn from $\mathcal{G}(n, 1/2)$. Write X_k as a sum of $\binom{n}{k}$ indicator variables, each corresponding to whether a particular subset of k vertices forms a clique. for a specific subset of k vertices, this happens with probability $2^{-\binom{k}{2}}$ (since each of the $\binom{k}{2}$ edges is included with probability $1/2$ in G). Applying linearity of expectation we see that

$$\mathbb{E}[X_k] = \binom{n}{k} 2^{-\binom{k}{2}}.$$

Using Markov's inequality (see Theorem .4 in Appendix A), we have

$$\Pr[X_k > 0] = \Pr[X_k \geq 1] \leq \mathbb{E}[X_k] = \binom{n}{k} 2^{-\binom{k}{2}}.$$

Thus, if k is such that $\binom{n}{k} 2^{-\binom{k}{2}} \to 0$ as $n \to \infty$, we clearly obtain that G almost surely does not have a clique of size k.

We expect that the value k_0 which achieves

$$\binom{n}{k_0} 2^{-\binom{k_0}{2}} = 1$$

is critical. We will prove that $k_0 = 2 \log n - 2 \log \log n + O(1)$. We need the following bound for $\binom{n}{k_0}$ which is proved in Appendix A.

$$\left(\frac{n}{k_0}\right)^{k_0} \leq \binom{n}{k_0} \leq n^{k_0} \leq \left(\frac{en}{k_0}\right)^{k_0}$$

Note that the above bounds are pretty good when $k = o(n)$. Hence we have that

$$\left(\frac{n}{k_0}\right)^{k_0} 2^{-\binom{k_0}{2}} \leq 1 \leq \left(\frac{en}{k_0}\right)^{k_0} 2^{-\binom{k_0}{2}},$$

which after some manipulations yields

$$2\log n + 1 \leq k_0 + 2\log k_0 \leq 2\log n + 2\log e + 1. \tag{1.1}$$

Now the monotonicity of $k_0 + 2\log k_0$ easily implies that $\log n < k_0 < 2\log n$ and consequently $\log\log n < \log k_0 < \log\log n + 1$. Plugging this into (1.1) we have that

$$2\log n - 2\log\log n - 1 < k_0 < 2\log n - 2\log\log n + 2\log e + 1$$

which proves that $k_0 = 2\log n - 2\log\log n + O(1)$.

Now let c_1, c_2 be some small integer constants (independent of n), which we will determine later. We will show the following

1. If $k > k_0 + c_1$ then $\mathbb{E}[X_k] \to 0$.
2. If $k < k_0 - c_2$ then $\mathbb{E}[X_k] \to \infty$.

For $k > k_0 + c_1$, we have that

$$\begin{aligned}
\binom{n}{k} 2^{-\binom{k}{2}} &\leq \left(\frac{en}{k}\right)^k 2^{-\binom{k}{2}} \\
&\leq e^k \left(\frac{n}{k_0}\right)^k \left(\frac{k_0}{k}\right)^k 2^{-\binom{k_0}{2}} 2^{-\frac{(k-k_0)(k+k_0-1)}{2}} \\
&\leq e^k \left(\frac{n}{k_0}\right)^{k_0} 2^{-\binom{k_0}{2}} \left(\frac{n}{k_0}\right)^{k-k_0} \left(\frac{k_0}{k}\right)^k 2^{-\frac{(k-k_0)(k+k_0-1)}{2}} \\
&\leq e^k \left(\frac{n}{k_0}\right)^{k-k_0} \left(\frac{k_0}{k}\right)^{k-k_0} 2^{-\frac{(k-k_0)(k+k_0-1)}{2}} \\
&\leq \left(\frac{n \cdot 2^{-\frac{k+k_0-1}{2}} e^{\frac{k}{k-k_0}}}{k}\right)^{k-k_0} \\
&\leq \left(\frac{n \cdot 2^{-\frac{k_0}{2}}}{k} \cdot 2^{-\frac{k-1}{2}} e^{\frac{k}{c_1}}\right)^{k-k_0}
\end{aligned}$$

Since $\frac{n \cdot 2^{-\frac{k_0}{2}}}{k} \leq 1$ for large enough n, it suffices to pick c_1 so that $2^{-\frac{k-1}{2}} e^{\frac{k}{c_1}} \to 0$. Some simple calculations show that $c_1 = 2.9$ is enough.

For $k < k_0 - c_2$, we have that

$$\binom{n}{k} 2^{-\binom{k}{2}} \geq \left(\frac{n}{k}\right)^k 2^{-\binom{k}{2}}.$$

It is easy to see that when $k \leq \log n$ this tends to infinity. Thus we assume that $\log n < k < k_0 - c_2$. Then

$$\begin{aligned}
\binom{n}{k} 2^{-\binom{k}{2}} &\geq \left(\frac{n}{k_0}\right)^k \left(\frac{k_0}{k}\right)^k 2^{-\binom{k_0}{2}} 2^{-\frac{(k-k_0)(k+k_0-1)}{2}} \\
&= \left(\frac{n}{k_0}\right)^{k_0} 2^{-\binom{k_0}{2}} \left(\frac{n}{k_0}\right)^{k-k_0} \left(\frac{k_0}{k}\right)^k 2^{-\frac{(k-k_0)(k+k_0-1)}{2}} \\
&= e^{-k_0} \left(\frac{en}{k_0}\right)^{k_0} 2^{-\binom{k_0}{2}} \left(\frac{n}{k_0}\right)^{k-k_0} \left(\frac{k_0}{k}\right)^k 2^{-\frac{(k-k_0)(k+k_0-1)}{2}} \\
&\geq e^{-k_0} \left(\frac{n}{k_0}\right)^{k-k_0} \left(\frac{k_0}{k}\right)^{k-k_0} 2^{-\frac{(k-k_0)(k+k_0-1)}{2}} \\
&\geq \left(\frac{k}{n \cdot 2^{-\frac{k_0}{2}}} \cdot 2^{\frac{k-1}{2}} e^{-\frac{k_0}{k_0-k}}\right)^{k_0-k} \\
&\geq \left(\frac{k}{n \cdot 2^{-\frac{k_0}{2}}} \cdot 2^{\frac{\log n - 1}{2}} e^{-\frac{2\log n}{c_2}}\right)^{k_0-k}
\end{aligned}$$

Since $\frac{k}{n \cdot 2^{-\frac{k_0}{2}}} \geq 1$, it suffices to pick c_2 so that $2^{-\frac{\log n - 1}{2}} e^{\frac{2\log n}{c_2}} \to \infty$. Some simple calculations show that $c_2 = 5.9$ is enough.

We have already proved that if $\mathbb{E}[X_k] \to 0$ then $\Pr[X_k > 0] \to 0$. Thus G has no clique of size $k > k_0 + c_1$.

We will now prove that for $k = k_0 - c_2$, $\Pr[X_k > 0] \to 1$. To do this, we will use the following inequality (whose proof can be found in Appendix A):

$$\Pr[X_k > 0] \geq \frac{(\mathbb{E}[X_k])^2}{\mathbb{E}[X_k^2]} \tag{1.2}$$

To apply this we first need to calculate $\mathbb{E}[X_k^2]$. Let us fix some labelling $\{1, \ldots, n\} = [n]$ of the vertices of G. Now, for $S \subseteq [n]$ so that $|S| = k$, let X_S denote the indicator variable which is 1 when the vertices S induce a clique (of size k) and 0 otherwise. Clearly

$$X_k = \sum_{S \subseteq [n], |S|=k} X_S$$

and consequently

$$X_k^2 = \sum_{S,T \subseteq [n]} X_S X_T = \sum_{S \subseteq [n]} X_S + \sum_{S \subseteq [n]} \sum_{T, S \neq T} \mathbb{E}[X_S X_T].$$

By linearity of expectation, we obtain that

$$\mathbb{E}[X_k^2] = \mathbb{E}[X_k] + \sum_{S\subseteq[n]} \sum_{T,T\neq S} \mathbb{E}[X_S X_T].$$

Since $\mathbb{E}[X_k] \to \infty$ for $k < k_0 - c_2$, to get the desired result from (1.2), it suffices to prove that

$$\frac{\sum_{S\subseteq[n]} \sum_{T,T\neq S} \mathbb{E}[X_S X_T]}{(\mathbb{E}[X_k])^2} \to 1 \text{ as } n \to \infty.$$

Note that

$$\begin{aligned}\sum_{S\subseteq[n]} \sum_{T,T\neq S} \mathbb{E}[X_S X_T] &= \sum_{S\subseteq[n]} \sum_{T,T\neq S} \mathbb{E}[X_T = 1 | X_S = 1]\Pr[X_S = 1] \\ &= \sum_{S\subseteq[n]} \Pr[X_S = 1] \sum_{T,T\neq S} \mathbb{E}[X_T = 1 | X_S = 1]\end{aligned}$$

and by symmetry $\sum_{T,T\neq S} \mathbb{E}[X_T = 1|X_S = 1]$ is the same for any S, so that

$$\begin{aligned}\sum_{S\subseteq[n]} \sum_{T,T\neq S} \mathbb{E}[X_S X_T] &= \left(\sum_{S\subseteq[n]} \Pr[X_S = 1]\right)\left(\sum_{T,T\neq S} \mathbb{E}[X_T = 1|X_S = 1]\right) \\ &= E[X_k] \sum_{T,T\neq S} \mathbb{E}[X_T = 1|X_S = 1].\end{aligned}$$

Hence it suffices to prove that

$$\frac{\sum_{T,T\neq S} \mathbb{E}[X_T = 1|X_S = 1]}{\mathbb{E}[X_k]} \to 1 \text{ as } n \to \infty.$$

Now it is easy to see that if $|S \cap T| = j$, we have that

$$\mathbb{E}[X_T = 1|X_S = 1] = \frac{1}{2^{\binom{k}{2}-\binom{j}{2}}} = 2^{-\binom{k}{2}+\binom{j}{2}}$$

Now, if we fix S, the number of sets T such that $|S \cap T| = j$ are $\binom{k}{j}\binom{n-k}{k-j}$ (this accounts for the number of ways to pick the j vertices in the intersection as well as the number of ways to pick the rest of the vertices in T). Thus, we obtain

$$\begin{aligned}\frac{\sum_{T,T\neq S} \mathbb{E}[X_T = 1|X_S = 1]}{\mathbb{E}[X_k]} &= \frac{\sum_{0\leq j\leq k-1} \binom{k}{j}\binom{n-k}{k-j} 2^{-\binom{k}{2}+\binom{j}{2}}}{\binom{n}{k} 2^{-\binom{k}{2}}} \\ &= \frac{\sum_{0\leq j\leq k-1} \binom{k}{j}\binom{n-k}{k-j} 2^{\binom{j}{2}}}{\binom{n}{k}}\end{aligned}$$

Let $a_j = \binom{k}{j}\binom{n-k}{k-j}2^{\binom{j}{2}}$ and set $f(j) = a_j/a_{j+1}$. After some manipulations, we have that

$$f(j) = \frac{(j+1)(n-2k+j+1)}{(k-j)^2 2^j}$$

Note that $f(2) >> 1$ and $f(k-1) << 1$. We will prove that $f'(j) < 0$ for $1 \leq j \leq k-3$, so that $a_j \leq \max\{a_1, a_{k-2}\}$ for $1 \leq j \leq k-2$. We have that

$$f'(j) = \frac{[2(j+1)+n-2k](k-j)^2 2^j}{(k-j)^4 2^{2j}} + \\ + \frac{(j+1)(n-2k+j+1)[2(k-j)2^j - (k-j)^2 2^j \ln 2]}{(k-j)^4 2^{2j}}.$$

Since $k - j > 0$, to prove that $f'(j) < 0$, it suffices that

$$[2(j+1)+n-2k](k-j) + (j+1)(n-2k+j+1)[2-(k-j)\ln 2] < 0$$

or equivalently

$$2 + \frac{n-2k}{j+1} < (n-2k+j+1)\left(\ln 2 - \frac{2}{k-j}\right).$$

For n large enough, it suffices to show that

$$\frac{1}{j+1} + \frac{2}{k-j} < \ln 2$$

Using that $\ln 2 \approx 0.693$, we can immediately check that this holds for n large enough (so that k is large enough) when $j \geq 1$ and $j \leq k-3$.

Hence, we have that

$$\frac{\sum_{T,T\neq S}\mathbb{E}[X_T = 1 | X_S = 1]}{\mathbb{E}[X_k]} = \frac{\sum_{0\leq j\leq k-1}\binom{k}{j}\binom{n-k}{k-j}2^{\binom{j}{2}}}{\binom{n}{k}} \\ = \frac{\binom{n-k}{k} + k(n-k)2^{\binom{k-1}{2}} + \sum_{1\leq j\leq k-2}\binom{k}{j}\binom{n-k}{k-j}2^{\binom{j}{2}}}{\binom{n}{k}}$$

Thus

$$\frac{\binom{n-k}{k}}{\binom{n}{k}} \leq \frac{\sum_{T,T\neq S}\mathbb{E}[X_T = 1 | X_S = 1]}{\mathbb{E}[X_k]} \tag{1.3}$$

and

$$\frac{\sum_{T,T\neq S}\mathbb{E}[X_T=1|X_S=1]}{\mathbb{E}[X_k]} \leq \frac{\binom{n-k}{k}}{\binom{n}{k}} + \frac{k(n-k)2^{\binom{k-1}{2}}}{\binom{n}{k}} + \\ + \frac{(k-2)\max\{k\binom{n-k}{k-1}, \binom{k}{2}\binom{n-k}{2}2^{\binom{k-2}{2}}\}}{\binom{n}{k}}. \quad (1.4)$$

Using (1.3) and (1.4), and the fact that k is roughly $2\log n$, it is easy to obtain

$$\frac{\sum_{T,T\neq S}\mathbb{E}[X_T=1|X_S=1]}{\mathbb{E}[X_k]} \to 1.$$

Thus we have proved that:

1. G does not have a clique of size $k_0 + c_1$ almost surely.
2. G almost surely has a clique of size $k_0 - c_2$.

Since $k_0 = 2\log n - 2\log\log n + O(1)$, this proves Theorem 1.1. ∎

As an aside of the proof of Theorem 1.1, note that we have actually proved that the maximum clique size in $G \in \mathcal{G}(n, 1/2)$ can take at most 10 different values! In fact, with a better analysis of the constants c_1, c_2 appearing in the proof, one can get the following tight and somewhat surprising result.

Theorem 1.2 ([BE76]). *The maximum clique in almost every graph $G \in \mathcal{G}(n, 1/2)$ has size either $k(n)$ or $k(n)+1$, for some integer $k(n)$.*

1.3 Finding Cliques in $\mathcal{G}(n, 1/2)$

Given the guarantees of Theorem 1.2, one might expect that we could find the maximum clique in $G \sim \mathcal{G}(n, 1/2)$. Unfortunately, this appears to be far away from reality. In fact, the following question has been open for a very long time.

Open Question. For any $\epsilon > 0$, construct a polynomial time algorithm that given as input a graph $G \sim \mathcal{G}(n, 1/2)$ output a clique of size $(1+\epsilon)\log n$ or prove that no one exists (modulo some standard complexity assumptions).

To get a feeling of the above question, it is worthy to mention that the natural greedy algorithm yields a clique of size $(1 + o(1))\log n$ and this

is the best we can do asymptotically. Thus, despite the fact that we can almost determine the size of the maximum clique, we cannot do better than a 2 approximation.

On the other hand, note that one can find the maximum clique in $G \sim \mathcal{G}(n, 1/2)$ by enumerating all possible subsets of size $\leq 2 \log n$. Hence, the problem can be solved in quasi polynomial time. Proving that no polynomial time algorithm exists would entail a reduction from worst case instances to average case instances and thus would probably be a highly non trivial construction. Of course this has been achieved for some problems, such as the shortest vector problem (see [Ajt98]).

Chapter 2

Planting a Clique

2.1 The distribution $\mathcal{G}(n, 1/2, k)$

Given the results of the previous chapter, it is interesting to examine whether planting a clique can help us recover the planted clique. In the model we are going to examine, a random n vertex graph is generated by randomly choosing k vertices to form a clique, and choosing every other pair of vertices independently with probability $1/2$ to be an edge. Denote the resulting distribution as $\mathcal{G}(n, 1/2, k)$. Our goal is to recover the set of k vertices which form the planted clique. We will refer to this problem as the planted clique problem.

Alternatively, we can generate $G \sim \mathcal{G}(n, 1/2, k)$ by first sampling a graph $G' \sim \mathcal{G}(n, 1/2)$. Then we choose a random subset of k vertices in G' and force them into a clique to obtain the final graph G. The distribution $\mathcal{G}(n, 1/2, k)$ was introduced independently by Jerrum ([Jer92]) and Kucera ([Kuc95]). Jerrum proved that the classical (by now) Metropolis process to perform a random walk on the set consisting of all cliques actually fails for $k = o(\sqrt{n})$, whereas Kucera displayed how to find such a clique when $k = \Omega(n \log n)$.

For the time being, we will not expand on the many research directions that have unfolded regarding the planted clique problem. Let us remark though that it has found extensions in cryptograhy and has interesting (and sometimes even unexpected) connections to other problems. Also, a good number of approaches have tried to find the minimum order of k for which the planted clique problem is solvable. We will discuss these approaches later, but less us spoil the discussion by saying that "all" we know is how to find a planted clique when $k = \Omega(\sqrt{n})$. This means, and compare this with the completely random case, that there is no hardness

proof for any $k = o(\sqrt{n})$. However, the large variety of techniques which have been applied in the quest of finding the minimum order of k is an exciting fact by itself.

Let us remark a few simple facts. First of all, note that for k large enough, say $k \geq 3\log n$, the planted clique is the maximum clique in a graph $G \sim \mathcal{G}(n, 1/2, k)$. To see this, we have already proved that the maximum clique in a completely random graph is roughly $2\log n$.

This suggests the following quasi polynomial time algorithm to solve the planted clique problem for $k \geq 3\log n$: try out all subsets of $3\log n$ vertices until you find one, say S, that forms a clique. Let C_S denote the common neighbours of vertices in S. Then, the planted clique is $S \cup C_S$ with high probability. To see this, note that by the observation above, the set S is with high probability a subset of the vertices which form the planted clique. Hence, the planted clique is included in the graph induced by $S \cup C_S$. It suffices to show that no vertex outside of the planted clique is included in $S \cup C_S$. But this is true, since for such a vertex the probability that it is connected to all vertices in S is $\left(\frac{1}{2}\right)^{3\log n} = \frac{1}{n^3}$. By a union bound over the $n - k$ vertices not in the clique, we obtain that, with probability $\geq 1 - \frac{1}{n^2}$, $S \cup C_S$ is exactly the planted clique.

This is also an indicator that the problem should become easier as k grows, which is also intuitive from a combinatorial viewpoint. In fact, the largest the gap between the planted solution and the generic solution, the easiest should be to recover the planted clique.

2.2 Planted Clique of size $\Omega(\sqrt{n\log n})$

In this section, we will display Kucera's algorithm ([Kuc95]) to find the planted clique when $k \geq \Omega(\sqrt{n\log n})$.

Let us first deal with the case $k \geq c\sqrt{n\log n}$ for some large enough constant c. The observation is that in this case the vertices of the clique are in fact the k vertices of larger degree in G. Assuming this, we can sort the vertices in decreasing degree order and just output the first k vertices.

To see why the aforementioned observation holds, let X_v denote the random variable which is equal to the degree of a vertex v in G. As we said in the previous section, we can sample the graph $G \sim \mathcal{G}(n, 1/2, k)$ as follows: first sample $G' \sim \mathcal{G}(n, 1/2)$ and afterwards plant a clique in a randomly chosen subset of k vertices.

We will prove that with high probability every vertex in G' has degree $n/2 \pm c'\sqrt{n\log n}$. To see this, write X_v as a sum of $n-1$ indicator variables (which indicate whether an edge is present or not) and note that by

Hoeffding's inequality (see Appendix A) we have

$$\Pr[|X_v - \mathbb{E}[X_v]| \geq c'\sqrt{n \log n}] \leq 2e^{-2c'^2 \log n} = 2\left(\frac{1}{n}\right)^{2c'^2}$$

Set c'=1. By a union bound over all n vertices, we see that with probability $\geq 1 - \frac{2}{n}$ every vertex in G' has degree $n/2 \pm \sqrt{n \log n}$. We claim that if we plant a clique of size roughly $4\sqrt{n \log n}$, the vertices of the clique will have the k highest degrees in G. Indeed, denote by P the planted clique and let $v \in P$. Using again a Hoeefding bound, we see that v has roughly $2\sqrt{n \log n}$ neighbors in G' inside P, so that after planting the clique it receives a degree boost of roughly $2\sqrt{n \log n}$. By a union bound over all vertices in P, we can prove that every vertex in P receives a degree boost of roughly $2\sqrt{n \log n}$. Since the lowest degree in G' is $n/2 - \sqrt{n \log n}$ and the highest is $n/2 + \sqrt{n \log n}$, the degree boost that vertices in P have received, made them the k highest degree vertices in G.

Thus we have proved the desired claim when $k \geq 4\sqrt{n \log n}$. But what if we just know that $k \geq \Omega(\sqrt{n \log n})$? Can we still find the clique? It is not hard to see that we could tighten our analysis and improve a little bit the constant c' (note that 4 is actually $4c'$ for $c' = 1$). But in fact we can use a slight trick (due to [AKS98]) and recover the clique in polynomial time and with enhanced probability of successs when $k \geq \Omega(\sqrt{n \log n})$. We will display this later in the chapter.

2.3 Planted Clique of size $\Omega(\sqrt{n})$

Note that the approach when $k \geq \Omega(\sqrt{n \log n})$ needed to look at the graph $G \sim \mathcal{G}(n, 1/2, k)$ only locally, i.e. it just needed the degree distribution of a verttex $v \in G$. To find planted cliques of asymptotically smaller sizes, we need to employ techniques that take into account global properties of the graph G.

There is a great variety of methods that achieve this. In this section, we will examine thoroughly the one which appears to be the most general and discuss briefly others in a later chapter.

2.3.1 The Spectral Approach

Alon, Krivelevich and Sudakov [AKS98] were the first to find an algorithm to recover a planted clique of size $k \geq \Omega(\sqrt{n})$. Our exposition of their result uses ideas culminating by a generalization of their approach due to McSherry ([McS01]). This exposition is sketched in [KV09].

To motivate the approach in [AKS98], recall that the first eigenvalue of a symmetric $n \times n$ matrix A is defined as

$$\lambda_1(A) = \max_{x \in \mathbb{R}^n, \|x\|=1} x^{\mathrm{T}} A x.$$

Denote by x_i the i-th coordinate of the vector $x \in \mathbb{R}^n$ and by A_{ij} the entry of the matrix A in the i-th row and j-th column. Note that

$$x^{\mathrm{T}} A x = \sum_{i,j} A_{ij} x_i x_j.$$

Now consider the $+1/-1$ version of the adjacency matrix of the graph $G = (V, E) \sim \mathcal{G}(n, 1/2, k)$. Namely let A be the (symmetric) matrix such that

$$A_{ij} = \begin{cases} 1, & \text{if } (i,j) \in E \\ 0, & \text{if } i = j \\ -1, & \text{if } (i,j) \notin E \end{cases}.$$

Let also x be the (normalized) indicator vector of the planted clique P (recall that $|P| = k$), namely

$$x_i = \begin{cases} \frac{1}{\sqrt{k}}, & \text{if } i \in P \\ 0, & \text{otherwise} \end{cases}.$$

Then, we have that

$$x^{\mathrm{T}} A x = \sum_{i,j} A_{ij} x_i x_j = 2\binom{k}{2}\frac{1}{k} \geq k - 1.$$

Thus, the planted clique forces the largest eigenvalue of the matrix A to be greater than $k - 1$. The crucial idea now is that if A was the $+1/-1$ version of the adjacency matrix of a graph $G' \sim \mathcal{G}(n, 1/2)$, then it is well known that its largest eigenvalue is less than $O(\sqrt{n})$. Thus for c sufficiently large and $k = c\sqrt{n}$, we can at first place distinct whether $G \sim \mathcal{G}(n, 1/2)$ or $G \sim \mathcal{G}(n, 1/2, k)$. Moreover, we can hope that the eigenvector corresponding to the largest eigenvalue of A will point to the planted clique. As we will see shortly, this is indeed the case.

Note that the above intuition relied on the fact that the largest eigenvalue of the $+1/-1$ version of the adjacency matrix of a graph $G' \sim G(n, 1/2)$ is $O(\sqrt{n})$. This is part of an important theorem which was first proved by Füredi and Komlós ([FK81]) and recently strengthened by Vu ([Vu05]). Vu also filled a minor gap in the original proof. We will not state their result but rather use the following weaker version of their theorem.

Theorem 2.1. *Suppose A is a symmetric $n \times n$ random matrix with independent above-diagonal entries uniformly distributed in ± 1. Then, with high probability, the largest eigenvalue of A is bounded by $4\sqrt{n}$.*

We will prove Theorem 2.1 at the end of the chapter. Our approach is different from those in [FK81] and [Vu05]. Both of these papers consider the value of $\text{Trace}((A-\mathbb{E}A)^r)$ and use it to bound $\lambda_1 \leq \text{Trace}((A-\mathbb{E}A)^r)^{1/r}$ for some large r. While their argument yields stricter results, we provide a probabilistic proof which has very nice ideas. We do not know where this proof first appeared, but it is a well known argument and it is sufficient for our purposes.

We have almost all the tools we need to display the algorithm that recovers a clique of size $k = c\sqrt{n}$ for some c large enough. The following simple lemma (which appeared in [McS01]) will be crucial for the proof. Let $\|A\|_2$ denote the 2-norm of a matrix A, i.e. $\|A\|_2 = \max_{\|x\|=1} \|Ax\|$. Note that for a symmetric matrix this is just the eigenvalue of largest absolute value. Finally, let $\|A\|_F$ denote the Frobenius norm of a matrix, i.e. the square root of the sum of squares of the entries of A. It is well known that if A has rank r, then $\|A\|_F \leq \sqrt{r}\,\|A\|_2$.

Lemma 2.2. *Suppose A, B are $m \times n$ matrices with $\text{rank}(B) = r$. If A_r is the best rank r approximation to A, then*

$$\|A_r - B\|_F^2 \leq 8r\,\|A - B\|_2^2$$

Proof Note that since $A_r - B$ has rank at most $2r$, we have that

$$\|A_r - B\|_F^2 \leq 2r\,\|A_r - B\|_2^2\,.$$

By triangle inequality, we have that $\|A_k - B\|_2 \leq \|A_k - A\|_2 + \|A - B\|_2$. Hence

$$\|A_r - B\|_F^2 \leq 2r(\|A_r - A\|_2 + \|A - B\|_2)^2$$

Since A_r is the best rank r approximation of A and since B has rank r, we have that $\|A_r - A\|_2 \leq \|A - B\|_2$ (see Appendix), so that

$$\|A_r - B\|_F^2 \leq 8r\,\|A - B\|_2^2$$ ∎

Note that if v denotes the eigenvector corresponing to the eigenvalue λ_1 with the largest magnitude, then the best rank 1 approximation is given by $A_1 = \lambda_1 v v^{\mathrm{T}}$.

We are now ready to state and prove the main theorem of this section.

Theorem 2.3. *Let $k \geq 46\sqrt{n}$. Then, there is a polynomial time algorithm which given a graph $G \sim \mathcal{G}(n, 1/2, k)$, outputs the vertices of the planted clique with high probability.*

Proof The algorithm we are going to analyze is the following:

1. Let A be the $+1/-1$ version of the adjacency matrix of the graph G.
2. Compute the eigenvector v corresponding to the largest eigenvalue of A.
3. Let S be the subset of k vertices of largest magnitude in v.
4. Output all the vertices that have at least $3k/4$ neighbors in S.

To prove its correctness, apply first Lemma 2.2 for $B = \mathbb{E}[A]$ and $r = 1$. We obtain

$$\|A_1 - \mathbb{E}[A]\|_F^2 \leq 8 \|A - \mathbb{E}[A]\|_2^2 .$$

By Theorem 2.1, this yieds

$$\|A_1 - \mathbb{E}[A]\|_F^2 \leq 128n.$$

Let $A^{(i)}$ denote the i-th column of the matrix A and note that

$$\|A_1 - \mathbb{E}[A]\|_F^2 = \sum_i \left(\left\| A_1^{(i)} - \mathbb{E}[A^{(i)}] \right\|_F^2 \right),$$

so that

$$\sum_i \left(\left\| A_1^{(i)} - \mathbb{E}[A^{(i)}] \right\|_F^2 \right) \leq 128n. \tag{2.1}$$

Let $\epsilon > 0$ to be determined later. By (2.1), for all but $\sqrt{n}/\epsilon$ columns the following holds:

$$\left\| A_1^{(i)} - \mathbb{E}[A^{(i)}] \right\|_F^2 \leq 128\epsilon\sqrt{n}. \tag{2.2}$$

Indeed, if this was not the case, then $\sum_i \left(\left\| A_1^{(i)} - \mathbb{E}[A^{(i)}] \right\|_F^2 \right) > 128n$ which contradicts (2.1).

Now let's pause for a second and think what a good handle (2.2) is. It says that the columns $A_1^{(i)}$ are close to the expectation columns $\mathbb{E}[A^{(i)}]$. Note that these columns are actually the same for each of the two "clusters", i.e. the planted clique and the rest of the graph. Namely, $\mathbb{E}[A^{(i)}]$

is the zero vector when i does not belong to the planted clique, and the indicator vector of the clique when i belongs to the clique. This suggests the following approach to partition the vertices of G: pick an arbitrary vertex i and find all vertices j whose corresponding column $A_1^{(j)}$ is close to $A_1^{(i)}$. Clearly if the indicator vector of the clique is sufficiently far from the zero vector, this approach will correctly classify the vertices in the clique and those not, at least for those vertices that (2.2) holds.

Of course, since (2.2) does not hold for all vertices, a refinement must be applied afterwards to fix wrongly classified vertices, but we will not dwelve deeper into the details. However, it is worth to mention that the above approach can be applied more generally, when the graph has more than two clusters, e.g. two planted cliques or a planted bipartition, etc. This is roughly the argument by McSherry in [McS01], which solves more generally the planted partition problem.

Now let's return to the analysis of the algorithm. The fact that we can just use the eigenvector corresponding to the largest eigenvalue to recover the planted clique (and avoid the above cluster method) is just a consequence of simple (but tricky) algebra.

Namely, recall that $A_1 = \lambda_1 v v^{\mathrm{T}}$ where λ_1 is the largest eigenvalue of A and v its corresponding eigenvector. Thus we have that $A_1^{(i)} = \lambda_1 v_i v$. Denote also by p the indicator vector of the clique (no normalization). Note that $\|p\|^2 = k$. Moreover,

$$\mathbb{E}[A_1^{(i)}] = \begin{cases} p, & \text{if } i \text{ belongs to the clique} \\ 0, & \text{otherwise} \end{cases}$$

For the rest of the proof, it would be better to try to reverse engineer the value of k for which the algorithm works correctly. For this to happen, the entries in v which correspond to vertices in the clique should be sufficiently dense in the k largest entries of v. We will use (2.2) to find the value of k for which this is indeed the case.

Let us first consider a vertex i which is not in the planted clique and for which (2.2) holds. Then $\left\|A_1^{(i)}\right\| = \lambda_1^2 v_i^2$ and $\mathbb{E}]A_1^{(i)}] = 0$, so that (2.2) yields

$$\lambda_1^2 v_i^2 \leq 128\epsilon\sqrt{n}. \tag{2.3}$$

Now consider a vertex i which is not in the planted clique and for which (2.2) holds. Then, use the inequality $\|a-b\|^2 \geq \frac{1}{2}\|b\|^2 - \|a\|^2$ (see Appendix for a proof), so that (2.2) yields

$$\frac{1}{2}k - 128\epsilon\sqrt{n} \leq \lambda_1^2 v_i^2. \tag{2.4}$$

Using (2.3), (2.4) we see that if $\frac{1}{2}k - 128\epsilon\sqrt{n} \geq 128\epsilon\sqrt{n}$, the entries of v within the clique will be greater than every other entry outside the clique (for those entries which (2.2) holds). The condition $\frac{1}{2}k - 128\epsilon\sqrt{n} \geq 128\epsilon\sqrt{n}$ implies that k should satify

$$k \geq 512\epsilon\sqrt{n}. \tag{2.5}$$

Now to make the whole argument work, we also need to make sure that the number of vertices such that (2.2) does not hold, say t, is sufficiently small compared to k, so that at least $k - t$ coordinates from the largest k coordinates of v point to the clique). Since the algorithm demands at least $3k/4$ coordinates to point to the clique, k should be such that $k - \frac{\sqrt{n}}{\epsilon} \geq 3k/4$, which yields

$$k \geq 4\frac{\sqrt{n}}{\epsilon}. \tag{2.6}$$

■

To satisfy (2.5) and (2.6) simultaneously, some simple algebra shows that the optimal choice for ϵ is $1/\sqrt{128}$, yielding $k \geq 46\sqrt{n}$.

To recapitulate, we have ensured that the set S recovered by the algorithm contains at least $3k/4$ vertices of the clique. By a Hoeffding bound and a union bound it is immediate to see that any vertex outside of the clique has at most $(2/3)k$ neighbors in S. Since vertices in the clique have at least $3k/4$ neighbors in S, the algorithm recovers exactly the planted clique with high probability.

By a slight tighter analysis, the condition $k \geq 46\sqrt{n}$ can be improved to $k \geq 20\sqrt{n}$. We omit the details.

2.4 Constant Improvements

In this section, we will show how to improve Theorem 2.3 to recover planted cliques of size $k = c\sqrt{n}$, for any constant c. We assume that we have an algorithm which finds planted cliques of size $k = c'\sqrt{n}$. The nice and elegant idea given by [AKS98] is the following:

1. Try to guess a constant number s of the vertices in the planted clique. Denote this set by S and by N_S the set of their common neighbors. Note that guessing means to pick a random subset of s vertices.

2. Run your algorithm on the graph induced by N_S.

3. Output $N_S \cup S$ iff $|N_S \cup S| = k$.

What is the expected running time of the above algorithm? Since a "guess" is correct with probability at least $(1/(c'\sqrt{n})^s)$, we have that after an expected number of roughly $O(n^{s/2})$ trials, we will find the desired subset S.

Suppose that at some point the algorithm indeed finds s vertices of the planted clique. Then the graph induced by N_S is a random graph on N_S vertices with a planted clique of size $n-s$. Note that here we used that the set S we guessed is a random subset of the initial planted clique. Note also that $|N_S| = \frac{n}{2^s}(1+o(1))$. To see this note that a vertex outside of the clique remains in the new graph with probality $1/2^s$. Thus, if s is chosen so that $c\sqrt{n} - s \geq c'\sqrt{n/2^s}$, our algorithm will correctly find the planted clique in the new graph (intuitively we have obtained a new instance of planted cique with improved parameters). This amounts to picking s roughly equal to $2\log(c'/c)$.

While the above technique is very general and works with almost all approaches, it has a major a drawback: the running time increases by a factor of $n^{2\log(c'/c)}$. In [DGGP10], the problem of improving the slow running time is posed.

2.5 Proof of Spectral Norm bound

In this section we prove Theorem 2.1. The proof uses the same steps as one given by Spielman.

Proof Let x be an arbitrary unit vector in $\mathbb{R}^n$ and consider the random variable $S = x^{\mathrm{T}}Ax$. We have that

$$S = 2\sum_{i<j} A_{ij}x_i x_j. \tag{2.7}$$

Since $\mathbb{E}[A_{ij}] = 0$, we have that $\mathbb{E}[S] = 0$. We are going to use Hoeffding's inequality (see Appendix) to bound the probability that S deviates more than t from its expectation. To do this, let us calculate the difference between the maximum and the minimum value of a term in (2.7). Since $|A_{ij}| \leq 1$, we have

$$-2x_i x_j \leq 2A_{ij}x_i x_j \leq 2x_i x_j,$$

so that the difference between the maximum and minimum value is $4x_ix_j$. Summing the squares of these differences, we have that

$$\sum_{i<j} 16x_i^2x_j^2 = 8\sum_{i,j} x_i^2x_j^2 = 8(\sum_i x_i^2)^2 = 8.$$

We can now apply Hoeffding's inequality to obtain that for an arbitrary vector $x \in \mathbb{R}^n$

$$\Pr[|x^{\mathrm{T}}Ax| \geq t] \leq 2e^{-\frac{t^2}{4}}. \tag{2.8}$$

While (2.8) is on the right path, it is not enough to obtain the desired result since x can be any vector in the unit ball. To overcome this obstacle we show that it is enough to look at a certain portion of the surface of the ball.

To see this, first observe that A, as it is a symmetric matrix, has n eigenvectors which form a basis of $\mathbb{R}^n$. Let $v^{(1)}, \ldots, v^{(n)}$ denote the eigenvalues of the matrix A ordered by the order of magnitude of their corresponding eigenvalues, so that the eigenvector corresponding to λ_1 is $v^{(1)}$. Take any x and decompose it as a linear combination of the $v^{(i)}$, i.e.

$$x = \sum_i \alpha_i v^{(i)} \text{ where } \sum_i \alpha_i^2 = 1.$$

Note that

$$\begin{aligned} x^{\mathrm{T}}Ax &= \sum_i \alpha_i^2\lambda_i \\ &\geq \alpha_1^2\lambda_1 - \lambda_1\left(\sum_{i\geq 2}\alpha_i^2\right) \\ &\geq \lambda_1(2\alpha_1^2 - 1) \end{aligned}$$

Hence, if $\alpha_1 \geq \sqrt{3}/2$, we obtain that $x^{\mathrm{T}}Ax \geq (1/2)\lambda_1$. Since $\alpha_1 = \left\langle x, v^{(1)}\right\rangle$, we obtain that

$$\text{if } x^{\mathrm{T}}v^{(1)} \geq \frac{\sqrt{3}}{2}, \text{ then } x^{\mathrm{T}}Ax \geq \frac{\lambda_1}{2}. \tag{2.9}$$

Now pick a random x in the unit ball. Then, by what we have just proved

$$\Pr_{x,A}\left[\lambda_1 \geq t \bigcap x^{\mathrm{T}}v^{(1)} \geq \frac{\sqrt{3}}{2}\right] \leq \Pr_{x,A}\left[x^{\mathrm{T}}Ax \geq \frac{t}{2}\right] \leq \Pr_A\left[x^{\mathrm{T}}Ax \geq \frac{t}{2}\right]. \tag{2.10}$$

Let v be an arbitrary unit vector. Then

$$\Pr_{x,A}\left[\lambda_1 \geq t \bigcap x^{\mathrm{T}} v^{(1)} \geq \frac{\sqrt{3}}{2}\right] = \Pr_{x,A}\left[\lambda_1 \geq t \bigcap x^{\mathrm{T}} v \geq \frac{\sqrt{3}}{2}\right] \quad (2.11)$$

$$= \Pr_x\left[x^{\mathrm{T}} v \geq \frac{\sqrt{3}}{2}\right] \Pr_A\left[\lambda_1 \geq t\right] \quad (2.12)$$

Note that (2.11) holds by the spherical symmetry of the distribution of a random unit vector. Combine (2.10), (2.12) to obtain

$$\Pr_A\left[x^{\mathrm{T}} A x \geq \frac{t}{2}\right] \geq \Pr_x\left[x^{\mathrm{T}} v \geq \frac{\sqrt{3}}{2}\right] \Pr_A\left[\lambda_1 \geq t\right]. \quad (2.13)$$

Thus it suffices to find $\Pr_x\left[x^{\mathrm{T}} v \geq \frac{\sqrt{3}}{2}\right]$ where v is an arbitrary unit vector. While we could compute this explicitly, we resort to a simple approximation argument. Let H be the hyperplane $x^{\mathrm{T}} v = \frac{\sqrt{3}}{2}$. This cuts a surface S' from the unit ball. It is clear that the required probability is the ratio of the area of S' over the area of the unit ball. Note that the area of S' is greater than the area of the $n-1$ dimensional ball defined by the restriction of H within the unit ball. The latter is a ball of radius $\sqrt{1-(\sqrt{3}/2)^2} = 1/2$. Using that the area of the n dimensional unit ball is

$$\frac{\pi^{\frac{n}{2}}}{\Gamma(\frac{n}{2}+1)},$$

we obtain that

$$\Pr_x\left[x^{\mathrm{T}} v \geq \frac{\sqrt{3}}{2}\right] \geq \frac{1}{\sqrt{\pi} n 2^{n-1}}$$

Plugging this back into (2.13), we have

$$\Pr_A\left[\lambda_1 \geq t\right] \leq \sqrt{\pi} n 2^{n-1} \cdot \Pr_A\left[x^{\mathrm{T}} A x \geq \frac{t}{2}\right] \leq \sqrt{\pi} n 2^n \cdot e^{-\frac{t^2}{16}}$$

Hence for $t > 4\sqrt{\ln 2}\sqrt{n} \approx 3.33\sqrt{n}$, the above probability becomes exponentially small which yields the desired result. ∎

There are many (and stricter) derivations of Theorem 2.1. [FK81] and [Vu05] prove it by combinatorial means. In [KV09] and [BV09], a lattice is used to discretize the space. Finally, [AKV02] makes use of Talagrand's inequality ([Tal96]).

Chapter 3

Extensions

While in the previous chapter we saw the best known achievable bounds for recovering planted cliques, there are other remarkable approaches that are worthy to mention.

3.1 Other Algorithmic Ideas

3.1.1 A Semidefinite Programming Approach

Feige and Krauthgamer [FK00] introduced a powerful method to find planted cliques based on the Lovász *theta* function. Their approach has two advantages:

1. Their algorithm works in the semirandom model as well. In the semirandom model, a graph G is first sampled from the distribution $\mathcal{G}(n, 1/2, k)$. Then, an adversary can remove edges which are not in the planted clique. Thus their algorithm has a nice robustness property that certain algorithms do not enjoy, e.g. Kucera's algorithm.

2. Their algorithm provides a certificate that the solution it outputs is optimal. This certificate is an upper bound for the clique number, which matches almost surely the output of their algorithm.

As mentioned before, their algorithm makes use of the Lovász *theta* function, usually denoted by $\theta(G)$ for a graph G. It is a relaxation of the independent set problem and can be computed using semidefinite programming to arbitrary precision ϵ in time poly$(\log \frac{1}{\epsilon})$.

Denote by $\bar{G}$, the complementary graph of G. The algorithm in [FK00] relies on the following lemma.

Lemma 3.1. *Let $G \in \mathcal{G}(n, 1/2, k)$, where $k > c\sqrt{n}$ for a sufficiently large constant c. Then $\theta(\bar{G}) = k$ with probability exponentially close to 1.*

The proof of Lemma 3.1 can be found in [FK00]. Using Lemma 3.1, an algorithm which recovers the clique and provides a certificate of optimality can be designed. Namely:

1. Let P be the set of vertices in G such that $\theta(\bar{G \setminus v}) \leq \theta(\bar{G}) - 1/2$.
2. Output $\theta(\bar{G})$ and P.

The algorithm can be modified so that it makes just one computation of the θ function.

To extend their algorithm to work for semirandom instances as well, Feige and Krauthgamer use the monotonicity of the θ function.

3.1.2 A Probabilistic Algorithm

The approaches we have seen up to now for the case $k = \Omega(\sqrt{n})$ are kind of non intuitive, at least at first sight. The following algorithm due to Feige and Ron ([FR10]) "fixes" this.

The algorithm consists of two phases. The first phase iteratively removes vertices of lowest degree until it ends up with a clique. The second phase expands the clique iteratively by checking which of the vertices which were removed in the first phase can be used to expand the clique.

The algorithm in [FR10], apart from being simple, has also the advantage that it runs in linear time. However, it has the disadvantage that it succeeds with constant probability (or at least this is what the authors prove). This was fixed by [DGGP10], where an algorithm is designed similar to the one we described above, but which refines the two phases.

3.1.3 The Tensor Approach

A very powerful approach for the planted clique algorithm was suggested by Frieze and Kannan ([FK08]). Their algorithm is based on 3-dimensional tensors, i.e. 3-dimensonal arrays.

Frieze and Kannan introduced the 3-dimensional parity tensor of a graph $G = (V, E)$, where the tensor"s entries denote the parity of the number of edges in subgraphs induced by 3 distinct vertices. More specifically, let E_{ij} equal to 1 if edge (i, j) is present in the graph G and -1 otherwise. Then, define the 3-dimensional tensor A with entries

$$A_{ijk} = \begin{cases} E_{ij}E_{ik}E_{jk} & \text{if } i \neq j, i \neq k, j \neq k \\ 0 & \text{otherwise} \end{cases}$$

We will denote by $\|A\|_2$ the 2-norm of the tensor A, i.e.

$$\|A\|_2 = \max_{\|x\|=1} A(x,x,x) = \max_{\|x\|=1} \sum_{i,j,k} A_{ijk} x_i x_j x_k$$

The idea they exploit is the same as in the case of matrices. Specifically, if we denote by k the size of the planted clique, then the (normalized) indicator vector of the clique forces the norm to be at least $\binom{k}{3} k^{-3/2} = \Omega(k^{3/2})$, whereas the 2-norm of the parity tensor corresponding to a random graph is roughly bounded by $\mathrm{O}(\sqrt{n})$ (modulo some logarithmic factors).

Their results can be summarized in the following two theorems.

Theorem 3.2. *There is a constant C such that with probability at least $1 - n^{-1}$ the norm of the 3-dimensional parity tensor $A : [n]^3 \to -1, 1$ for the random graph $G(n, 1/2)$ is bounded by*

$$\|A\|_2 \leq C\sqrt{n} \log^4 n$$

Theorem 3.3. *Let x be a vector such that $A(x,x,x) \geq \alpha^r \|A\|_2$. Then, for p such that*

$$n \geq p > C\alpha^{-2} n^{1/3} \log^3 n$$

the planted clique can be recovered with high probability in polynomial time.

The proof of Theorem 3.2 by [FK08] is combinatorial. Their approach was simplified and generalized to r-dimensional tensors by Brubaker and Vempala ([BV09]). They prove that a clique of size roughly equal to $n^{1/r}$ can be recovered if one can maximize (or even approximate within some factors) the tensor norm. Their proof yields an analogous theorem to Theorem 3.2.

Do their approaches achieve breaking the $o(\sqrt{n})$ barrier? As it was proved later, no. It turned out that maximizing 3-dimensional tensor norms is generally NP-hard ([HL09]). Still, it is not known whether this extends to the the special structure (and randomness) that A has.

3.2 Connections to Other Problems

As we have seen, it seems that it is difficult to cross the barrier of finding a planted clique of size $o(\sqrt{n})$. This naturally lead to the conjecture (which is still open) that finding a planted clique of sufficiently small size is a computationally hard problem. The exact formulation of the conjecture is the following.

Conjecture. Finding a planted clique of size $k = O(\log n)$ is hard.

We briefly present some results which demonstrate the rich depth of the planted clique problem.

3.2.1 Cryptography

Juels and Peinado, assuming the conjecture, use the planted clique to construct several cryptographic protocols ([JP00]).

One of them is used to create hierarchical keys. In this setting, there are t parties $P_1, \ldots, P_t$. Assume for simplicity that $t = O(1)$. We want to assign a key to each party so that P_i knows all the keys of the parties P_j with $j > i$ (but not vice versa). The protocol which is proposed in [JP00] is the following:

1. P_1 samples a graph G from the distribution $\mathcal{G}(n, 1/2)$.
2. P_1 plants a clique of size $O(\log n)$ and then passes the new graph to P_2.
3. P_2 plants a clique of size $O(\log n)$ and then passes the new graph to P_3, and so on.
4. P_t plants a clique of size $O(\log n)$ and then publishes the final graph.

The key corresponding to each player is the clique he planted. Clearly, the cliques which the parties planted are with high probability disjoint. Note also that each player P_i knows what the graph was before any player P_j with $j > i$ planted his own clique and consequently P_i can recover the keys of the players P_j with $j > i$.

On the other hand, if finding a planted clique is computationally hard, then no party P_j can recover the key of a party P_i with $i < j$.

3.2.2 Complexity

Hazan and Krauthgamer ([HK09]) proved that if there is a polynomial time algorithm which are ϵ-close to the best Nash equilibrium of a 2 player game which maximizes social welfare (the sum of players' payoffs), then a planted clique of size $O(\log n)$.

Note that as in the case of planted clique, there is an $n^{O(\log n)}$ algorithm to find approximate Nash equilibria due to Lipton, Markakis and Mehta ([LMM03]).

Finally, Alon et al. ([AAK$^+$07]) proved that if there is a polynomial time algorithm which can test ϵ-closeness to $O(\log n)$-pairwise independence given polynomial number of samples, then one can recover a planted clique of size $O(\log n)$.

References

[AAK+07] Noga Alon, Alexandr Andoni, Tali Kaufman, Kevin Matulef, Ronitt Rubinfeld, and Ning Xie. Testing k-wise and almost k-wise independence. In *STOC*, pages 496–505, 2007.

[Ajt98] Miklós Ajtai. The shortest vector problem in $_2$ is p-hard for randomized reductions (extended abstract). In *STOC*, pages 10–19, 1998.

[AKS98] Noga Alon, Michael Krivelevich, and Benny Sudakov. Finding a large hidden clique in a random graph. In *SODA*, pages 594–598, 1998.

[AKV02] Noga Alon, Michael Krivelevich, and Van H. Vu. On the concentration of eigenvalues of random symmetric matrices. *Israel Journal of Mathematics*, (131):259b•"267, 2002.

[BE76] Béla Bollobás and Paul Erdös. Cliques in random graphs. *Mathematical Proceedings of the Cambridge Philosophical Society*, 80, 1976.

[BV09] S. Charles Brubaker and Santosh Vempala. Random tensors and planted cliques. *CoRR*, abs/0905.2381, 2009.

[DGGP10] Yael Dekel, Ori Gurel-Gurevich, and Yuval Peres. Finding hidden cliques in linear time with high probability. *CoRR*, abs/1010.2997, 2010.

[FK81] Zoltán Füredi and János Komlós. The eigenvalues of random symmetric matrices. *Combinatorica*, 1(3):233–241, 1981.

[FK00] Uriel Feige and Robert Krauthgamer. Finding and certifying a large hidden clique in a semirandom graph. *Random Struct. Algorithms*, 16(2):195–208, 2000.

[FK08] Alan M. Frieze and Ravi Kannan. A new approach to the planted clique problem. In *FSTTCS*, pages 187–198, 2008.

[FR10] Uriel Feige and Dana Ron. Finding hidden cliques in linear time. In *AOFA*, 2010.

[Hås97] Johan Håstad. Some optimal inapproximability results. *Electronic Colloquium on Computational Complexity (ECCC)*, 4(37), 1997.

[HK09] Elad Hazan and Robert Krauthgamer. How hard is it to approximate the best nash equilibrium? In *SODA*, pages 720–727, 2009.

[HL09] Christopher Hillar and Lek-Heng Lim. Most tensor problems are np hard. *CoRR*, abs/0911.1393, 2009.

[Jer92] Mark Jerrum. Large cliques elude the metropolis process. *Random Struct. Algorithms*, 3(4):347–360, 1992.

[JP00] Ari Juels and Marcus Peinado. Hiding cliques for cryptographic security. *Des. Codes Cryptography*, 20(3):269–280, 2000.

[Kar72] Richard M. Karp. Reducibility among combinatorial problems. In R. E. Miller and J. W. Thatcher, editors, *Complexity of Computer Computations*, pages 85–103. Plenum Press, 1972.

[Kuc95] Ludek Kucera. Expected complexity of graph partitioning problems. *Discrete Applied Mathematics*, 57(2-3):193–212, 1995.

[KV09] Ravi Kannan and Santosh Vempala. Spectral algorithms. *Foundations and Trends in Theoretical Computer Science*, 4(3-4):157–288, 2009.

[LMM03] Richard J. Lipton, Evangelos Markakis, and Aranyak Mehta. Playing large games using simple strategies. In *ACM Conference on Electronic Commerce*, pages 36–41, 2003.

[McS01] Frank McSherry. Spectral partitioning of random graphs. In *FOCS*, pages 529–537, 2001.

[MR95] Rajeev Motwani and Prabhakar Raghavan. *Randomized Algorithms*. Cambridge University Press, 1995.

[Tal96] Michel Talgrand. Concentration of measure and isoperimetric inequalities in product spaces. *Publications I.H.E.S*, (81):73–205, 1996.

[Vu05] Van H. Vu. Spectral norm of random matrices. In *STOC*, pages 423–430, 2005.

[Zuc06] David Zuckerman. Linear degree extractors and the inapproximability of max clique and chromatic number. In *STOC*, pages 681–690, 2006.

Appendices

Useful Inequalities

Theorem .4 (Markov's Inequality). *Let X be a random variable which takes only positive values and has bounded expectation. Then for any real number $t > 0$,*

$$\Pr[X \geq t] \leq \frac{\mathbb{E}[X]}{t}$$

Proof Straightforward. Since X takes only positive values, by the definition of expectation, we have

$$\mathbb{E}[X] \geq t\Pr[X \geq t].$$ ∎

Theorem .5 (Chebyshev's Inequality). *Let X be a random variable with expected value μ and variance σ^2. Then for any real number $k > 0$,*

$$\Pr[|X - \mu| \geq k\sigma) \leq \frac{1}{k^2}$$

Proof Define

$$g(x) = \begin{cases} 1, & \text{if } |X - \mu| \geq k\sigma \\ 0, & \text{otherwise} \end{cases}$$

Note that $0 \leq g(x) \leq \dfrac{(X-\mu)^2}{k\sigma}$. Then

$$\Pr[|X - \mu| \geq k\sigma) = \mathbb{E}[g(x)] \leq \mathbb{E}[\frac{(X-\mu)^2}{k\sigma}] = \frac{1}{k^2\sigma^2}\mathbb{E}[(X-\mu)^2]$$

The desired inequality follows since

$$\mathbb{E}[(X-\mu)^2] = \mathbb{E}[X^2 - 2\mu X + \mu^2] = \mathbb{E}[X^2] - \mu^2 = \sigma^2$$ ∎

Theorem .6 (The Second Moment method). *Let X be a random variable which takes only nonnegative values. Then*

$$\Pr[X > 0] \geq \frac{(\mathbb{E}[X])^2}{\mathbb{E}[X^2]}$$

Proof Define

$$g(x) = \begin{cases} 1, & \text{if } X > 0 \\ 0, & \text{otherwise} \end{cases}$$

Then using Cauchy-Schwarz inequality we have that

$$\mathbb{E}[X] = \mathbb{E}[X \cdot g(X)] \leq \sqrt{\mathbb{E}[X^2]} \cdot \sqrt{\mathbb{E}[g(X)^2]}$$

The inequality follows since $\mathbb{E}[g(X)^2] = \Pr[X > 0]$. ∎

Theorem .7 (Hoeffding's Inequality). *Let $X = X_1 + \ldots + X_n$ where the X_i"s are independent random variables. If $a_i \leq X_i - \mathbb{E}[X_i] \leq b_i$ for every $1 \leq i \leq n$, then for every $t > 0$, we have that*

$$\Pr[|X - \mathbb{E}[X]| \geq t] \leq 2e^{-\frac{2t^2}{\sum_i (b_i - a_i)^2}}$$

Proof Fix $t > 0$, apply Markov's inequality on the random variable e^{tX} and then optimize t. The details of the proof can be found in many sources. One of them is [MR95]. ∎

Proposition .8 (Bound for Binomial Coefficients). *For every $0 < k < n$, the following inequality holds:*

$$\left(\frac{n}{k}\right)^k \leq \binom{n}{k} \leq \left(\frac{en}{k}\right)^k$$

Proof For the left part, use that $\binom{n}{k} = \frac{n}{k}\binom{n-1}{k-1}$ and apply induction.

For the right part, note that

$$\binom{n}{k} = \frac{n(n-1)\cdots(n-k)}{k!} \leq \frac{n^k}{k!}$$

Thus it suffices to prove that

$$\frac{k^k}{k!} \leq e^k,$$

which follows easily from the series expansion of e^x. ∎

Theorem .9. *For any matrix B with rank r and any matrix A, it holds that $\|A - A_r\|_2 \leq \|A - B\|_2$*

Proof Let $u^{(1)}, \ldots, u^{(r)}, u^{(r+1)}$ denote the top $r+1$ left singular vectors of A (if rank of A is smaller than $r+1$ the inequality holds trivially since $A_r = A$). Since B is a rank r matrix and the vectors $u^{(i)}$ are orthonormal to each other, we have that for some $1 \leq i \leq r+1$, $u^{(i)}B = 0$. For this i, we obtain that $\left\|u^{(i)}(A - B)\right\| = \left\|u^{(i)}A\right\| \geq \|A - A_r\|_2$ since $u^{(i)}$ is one of the top $r+1$ left singular vectors of A. ∎

Proposition .10. *For any two vectors $a, b \in \mathbb{R}^n$ the following inequality holds:*

$$\frac{1}{2}\|b\|^2 - \|a\|^2 \leq \|a - b\|^2 .$$

Proof By expanding the norms, it suffices to prove the inequality for arbitrary real numbers a, b. This amounts to proving

$$\frac{1}{2}b^2 - a^2 \leq (a - b)^2,$$

which is equivalent to

$$0 \leq (2a - b)^2.$$

■