Random Graphs — Week 7

Yuval Filmus

December 6, 2019

1 Finding a clique in $G(n, 1/2)$

Last week we saw that with high probability, $\omega(G(n,1/2)) \approx 2\log_2 n$ (recall that $\omega(G)$ is the size of the maximum clique in G; similarly, $\alpha(G)$ is the size of the maximum independent set in G). Can we find such a clique?

Going over all sets of $2\log_2 n$ vertices, we can find such a clique in quasipolynomial time $n^{O(\log n)}$. But can we do it in polynomial time?

The following algorithm comes to mind. Start with an arbitrary vertex. Remove all of its non-neighbors, and repeat. How well does this algorithm perform? Intuitively, each vertex we add cuts the number of vertices by one half, so the algorithm should terminate after roughly $\log_2 n$ steps.

Indeed, let N_i be the number of vertices which are connected to the first i vertices of the clique (if there are less than i vertices in the clique, we define $N_i = 0$). Then $N_0 = n$, and

$$\mathbb{E}[N_{i+1} \mid N_i] = \begin{cases} \frac{N_i - 1}{2} & \text{if } N_i \geq 1, \\ 0 & \text{otherwise.} \end{cases}$$

In both cases, $\mathbb{E}[N_{i+1} \mid N_i] \leq N_i/2$, and so a simple induction shows that

$$\mathbb{E}[N_k] \leq \frac{n}{2^k}.$$

Let p_k be the probability that the algorithm constructs a clique of size at least k. If this happens, then $N_{k-1} \geq 1$, and so

$$p_k = \Pr[N_{k-1} \geq 1] \leq \frac{n}{2^{k-1}}.$$

In particular, if $k = \log_2 n + C$, then $p_k \leq 1/2^{C-1}$. Hence if $C \to \infty$, then with high probability the algorithm constructs a clique of size less than $\log_2 n + C$.

Proving a lower bound requires using the second moment method, which is more difficult. We will use an elegant analysis found in lecture notes of Luca Trevisan.

Here is another way to consider the algorithm. Fix some arbitrary order $v_1, \ldots, v_n$ of the vertices. Go over them one by one, and add to the clique any vertex which is connected to the previous vertices in the clique. Let $i_1, i_2, \ldots$ be the indices of vertices added to the clique. Then $i_1 = 1$, and $i_2 - i_1$ has roughly a geometric distribution $G(1/2)$

— with the caveat that we could run out of vertices. Similarly, $i_3 - i_2$ has roughly a geometric distribution $G(1/4)$, and so on.

The problem with the idea presented above is that we could run out of vertices. This is however easy to fix: we just use infinitely many vertices, only the first n of which are "real"! Now $i_{\ell+1} - i_\ell \sim G(2^{-\ell})$ (where $i_0 = 0$). The size of the clique is the maximum k such that $i_k \leq n$.

We can calculate $\mathbb{E}[i_{\ell+1} - i_\ell] = 2^\ell$, and so

$$\mathbb{E}[i_k] = \sum_{\ell=0}^{k-1} 2^\ell = 2^k - 1.$$

Let $k = \log_2 n - C$. Then

$$\Pr[i_k > n] < \frac{\mathbb{E}[i_k]}{n} < \frac{n/2^C}{n} = 2^{-C}$$

It follows that if $C = \omega(1)$, then with high probability $i_k \leq n$, that is, the algorithm finds a clique of size at least $\log_2 n - C$.

It is conjectured that no efficient algorithm can find a clique of size $(1+\epsilon)\log_2 n$ in $G(n, 1/2)$, even with constant success probability.

2 Planted clique

Finding the maximum clique in $G(n, 1/2)$ seems hard. What if we force the graph to contain a larger clique? Does it make it easier to find the clique? The standard model in this case is $G(n, 1/2, k)$, in which we choose a random k-clique, and then put in every other edge with probability $1/2$.

First, let us verify that unless k is very small, $G(n, 1/2, k)$ only contains a single k-clique (with high probability).

Lemma 1. *If $k = \omega(\log n \log\log n)$, then with high probability, $G(n, 1/2, k)$ contains a single k-clique. In particular, $\omega(G(n, 1/2, k)) = k$ with high probability.*

Proof. With high probability, the vertices outside the planted clique do not support a clique of size $\log_2(n-k) \leq \log_2 n$.

Assuming this, any k-clique different from the planted clique must contain a vertex connected to at least $k - \log_2 n$ vertices from the planted clique. This happens with probability at most

$$n\binom{k}{\log_2 n} 2^{-(k-\log_2 n)} \leq \frac{n^2 k^{\log_2 n}}{2^k} = 2^{(2+\log_2 k)\log_2 n - k} = o(1). \qquad \square$$

2.1 Kučera's algorithm

Perhaps the simplest observation is that the vertices of the planted clique have higher degree than non-clique vertices. This is because the degree of a normal vertex is distributed $\mathrm{Bin}(n-1, 1/2)$, whereas the degree of a vertex belonging to the planted clique is distributed $k - 1 + \mathrm{Bin}(n-k, 1/2)$, with expectation $\frac{n-k}{2} + k - 1 = \frac{n-1}{2} + \frac{k-1}{2}$.

How large should k be so that we are able to distinguish between these two distributions? To answer this question, we need to appeal to a large deviation bound such as the Chernoff bound. One formulation of this bound (called Hoeffding's inequality) is as follows:

$$\Pr[\mathrm{Bin}(m,p) \leq (p-\epsilon)m], \Pr[\mathrm{Bin}(m,p) \geq (p+\epsilon)m] \leq e^{-2\epsilon^2 m}.$$

The degree of a normal vertex has distribution roughly $\mathrm{Bin}(n,1/2)$, and there are roughly n such vertices. Chernoff's bound predicts that the maximal degree is around $n/2+\epsilon n$, where $e^{-2\epsilon^2 n} = 1/n$, that is, $2\epsilon^2 n = \log n$, which implies that $\epsilon = \sqrt{\frac{\log n}{2n}}$, and so $\epsilon n = \sqrt{\frac{1}{2} n \log n}$. This means that for the algorithm to have any chance of success, we need $\frac{k-1}{2} \geq \sqrt{\frac{1}{2} n \log n}$, that is, $k \geq C\sqrt{n \log n}$ for some appropriate $C > 0$.

Conversely, the Chernoff bound shows that with high probability, all the degrees of normal vertices are at most (say) $n/2 + \sqrt{n \log n}$, whereas all the degrees of the planted clique vertices are at least (say) $n/2 + k/2 - \sqrt{n \log k}$. In particular, for large enough C, we can find the planted clique simply by taking the k vertices of maximal degree.

2.2 Partial enumeration

Using a simple idea, we can replace the constant C by any other constant C_0. The idea is to "guess" a few vertices of the clique. Suppose that somebody revealed to us ℓ vertices from the clique. Looking only at their common neighbors, we reduce the number of vertices from ℓ to roughly $k - \ell + \frac{n-k}{2^\ell} \approx n/2^\ell$, while reducing the planted clique from k to $k-\ell$. In effect, we have increased the value of C_0 by a factor of (almost) 2^ℓ. Choosing $\ell = \log_2(C/C_0)$, we can apply Kučera's algorithm.

In practice, nobody is going to give us ℓ vertices of the clique. However, we can go over all subsets of ℓ vertices, increasing the running time by a factor of $O(n^\ell)$. For the correct choice of ℓ vertices, we will manage to find the planted clique. (Recall that with high probability, $G(n,1/2,k)$ contains a unique k-clique, and so we cannot find the "wrong" clique.)