Assignment 2: Solution Sketch

Itay Hazan
Random Graphs, Technion, 2017

January 19, 2018

1 Solving NP-Hard Problems on Random Graphs

First, we make the following important observations:

- Since $\frac{c}{n} = \frac{1}{n} - \omega\left(\frac{1}{n^{4/3}}\right)$, then whp every component of the graph is either a tree or unicyclic (see the section on the Unicyclic Regime in the lecture notes).
- For any $f(n) = \omega(1)$, the size of the largest component of $G(n, c/n)$ is at most $O(\log n + f(n))$. Choosing $f(n) = \log\log n$ shows that the largest component contains $O(\log n)$ vertices (whp).

We now sketch an algorithm and a brief correctness argument for each of the mentioned problems:

Maximum Clique. Since the graph is unicyclic whp, then there are no cliques on four vertices or more in the graph (whp). It therefore suffices to find a clique on three vertices or less in order to output a maximum clique whp. The simplest brute force algorithm, that goes over all possible subsets of V of size 3 or less and checks if they define a clique will do.

Maximum Independent Set. The largest independent set of a graph is the union of the largest independent sets of each of its components. It therefore suffices to argue a way to find the maximum independent set of each component. Since we know that each component contains at most $c \log n$ vertices whp, then a brute force algorithm that finds a maximum independent set in a component by going through all possible subsets of the vertices and checking for each subset whether it forms an independent set takes $2^{c \log n} = n^c = \text{poly}(n)$. Since there are at most n components, then applying this procedure for every component takes polynomial time.

Minimum Vertex Cover. Recall that for every graph $G = (V, E)$, $S \subset V$ is an independent set of G iff $V \setminus S$ is a vertex cover for G. Therefore, if S is a maximum independent set the $V \setminus S$ is a minimum vertex cover. We can therefore run the algorithm for finding (whp) a maximum independent set, and returning the complement of its output.

Minimum Feedback Vertex Set. Given a graph G, we must remove at least one vertex from every cycle in G in order to transform it into a tree. Since whp every component in $G(n, c/n)$ is unicyclic, then it suffices to run DFS on each component and remove every vertex from which a backwards edge goes out.

Chromatic Number. Recall that in $G(n, c/n)$, each component is either a tree or unicyclic whp. We know that the chromatic number of trees is 2, and the chromatic number of cycles is either 2 or 3, depending on the parity of the number of vertices: if there is an even number of vertices, then the cycle can be colored with two alternating colors, and otherwise we need an extra color. We therefore output the following: if there is an odd cycle in G, answer '3'. Otherwise, answer '2'.

2 Unicyclic Graphs, Empirically

2.a. Let $N = \binom{\binom{n}{2}}{n}$ denote the number of graphs with n vertices and n edges. Denote $p = \Pr[G(n,n) \text{ is connected}]$. Clearly,

$$p = \frac{U_n}{N}.$$

2.b. Suppose we sample t graphs $G_1, \ldots, G_t \sim G(n,n)$, and let us define an indicator I_i for every $i \in [t]$ as follows:

$$I_i = \begin{cases} 1 & \text{if } G_i \text{ is connected} \\ 0 & \text{otherwise} \end{cases}.$$

Clearly, $I_1, \ldots, I_t$ are i.i.d. $\mathrm{Ber}(p)$ random variables. Define $\bar{I} = \frac{1}{t}\sum_{i\in[t]} I_i$. By the weak law of large numbers, $\bar{I} \xrightarrow{P} p$, and therefore we can use $\bar{I}$ as an estimator for p. We can therefore estimate U_n by $\left[\bar{I} \cdot \binom{\binom{n}{2}}{n}\right]$.

2.c. Since the I_is are i.i.d., then

$$\mathbb{V}[\sum_{i\in[t]} I_i] = \sum_{i\in[t]} \mathbb{V}[I_i] = tp(1-p).$$

Therefore,

$$\mathbb{V}[\bar{I}] = \mathbb{V}\left[\frac{1}{t}\sum_{i\in[t]} I_i\right] = \frac{1}{t^2}\mathbb{V}[\sum_{i\in[t]} I_i] = \frac{p(1-p)}{t}.$$

Since $X = \left[\bar{I} \cdot \binom{\binom{n}{2}}{n}\right]$, then

$$\mathbb{V}[X] = \binom{\binom{n}{2}}{n}^2 \cdot \frac{p(1-p)}{t},$$

i.e.

$$\sigma[X] = \binom{\binom{n}{2}}{n} \cdot \sqrt{\frac{p(1-p)}{t}}.$$

2.d. By the *68-95-99.7 rule*, $\Pr[|X - U_n| \leq 3\sigma(X)] \approx 0.997$. Requiring $\Pr[X = U_n] \approx 0.997$ is the same as requiring $\Pr[|X - U_n| < 1/2] \approx 0.997$. It therefore suffices to demand that $3\sigma(X) < \frac{1}{2}$, namely:

$$t > 36\binom{\binom{n}{2}}{n}^2 \cdot p(1-p).$$

It therefore also suffices to require

$$t > 36\binom{\binom{n}{2}}{n}^2 \cdot p,$$

and since $U_n = p \cdot \binom{\binom{n}{2}}{n}$, then we get:

$$t > 36\binom{\binom{n}{2}}{n} U_n \approx 36\binom{\binom{n}{2}}{n} n^n$$

.

2.e. I wrote the following code, that uses a python library called *networkx* to easily represent graphs:

```
import networkx as nx
from scipy.special import binom

for n in [3,4,5]:
    num_graphs = int(36 * binom(binom(n,2),n) * n**n) + 1
    num_connected = 0
    for i in range(num_graphs):
        num_connected += int(nx.is_connected(nx.gnm_random_graph(n,n)))

    print('U_{} = {}'.format(n, num_connected/num_graphs*binom(binom(n,2),n)))
```

The output was:

```
U_3 = 1.0
U_4 = 15.0
U_5 = 222.005
```

3 Unicyclic Graphs, Combinatorially

3.a. Every connected unicyclic graph can be viewed as a cycle whose vertices are "roots" for trees.

Let $3 \leq k \leq n$ denote the number of vertices in the cycle. We sum over the possible values of k: There are two main approaches towards this:

1. The number of rooted k-forests on n vertices is $\binom{n}{k} k n^{n-k-1}$ (see *Cayley's Formula: A Page from a Book* by Arnon Avron and Nachum Dershowitz for proof). After partitioning the n vertices into k rooted trees, we connect all of the roots in a cycle. There are $\frac{(k-1)!}{2}$ ways of sorting k elements in a cycle, and hence we get:

$$U_n = \sum_{k=3}^{n} \binom{n}{k} k n^{n-k-1} \cdot \frac{(k-1)!}{2}.$$

2. First, choose k vertices for the cycle $\binom{n}{k}$, order them $\frac{(k-1)!}{2}$, and then partition the rest of the vertices into k sets for the k trees $\binom{n-k}{n_1,\ldots,n_k}$, and order them within the trees.

$$U_n = \sum_{k=3}^{n} \frac{(k-1)!}{2} \binom{n}{k} \sum_{\substack{0 \leq n_1,\ldots,n_k \leq n-k \\ \sum n_i = n-k}} \binom{n-k}{n_1,\ldots,n_k} \prod_{i=1}^{k} (n_i+1)^{n_i-1}.$$

3.b. The calculations are left for the reader.

4 Unicyclic Graphs, Using the Internet

Plugging '1, 15, 222, 3360' into the *Online Encyclopedia of Integer Sequences* gives the following page. In one of the results in this page, the following asymptotic expansion of U_n is given:

$$U_n = n^{n-\frac{1}{2}} \left(\frac{\sqrt{2\pi}}{4} - o(1) \right).$$

We therefore conclude that $U_n = \Theta(n^{n-\frac{1}{2}})$.