Probabilistic Method and Random Graphs

Lecture 9. Random Graphs-Part II[1]

Xingwu Liu

Institute of Computing Technology
Chinese Academy of Sciences, Beijing, China

[1] Mainly based on Lecture 13 of Ryan O'Donnell's lecture notes of *Probability and Computing. The application is based on Chapter 5.6 in Probability and Computing*.

Questions, comments, or suggestions?

Recap of Lecture 8

Random graphs are motivated by modeling gigantic graphs

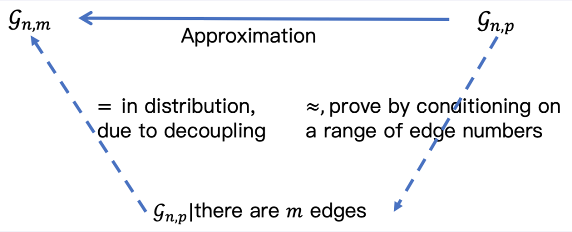
Two views of random graphs

- Probability space over graphs
  - Equal probability on all n-graphs: $\mathcal{G}_n$
  - Equal probability on all n-graphs with m edges: $\mathcal{G}_{n,m}$
  - Hard to compute statistics
- Generated by stochastic processes
  - Play a super dice
  - Determine each edge by independently tossing a coin: $\mathcal{G}_{n,p}$
    - $\mathcal{G}_{n,\frac{1}{2}} \sim \mathcal{G}_n$, easy to compute statistics
    - A spectrum of probability spaces on the same sample space
  - Independently randomly sample m edges: $\mathcal{G}_{n,m}$

Recap of Lecture 8

Decoupling dependency in $\mathcal{G}_{n,m}$

- $\mathcal{G}_{n,m} \sim (\mathcal{G}_{n,p} | \text{there are } m \text{ edges})$
- A paradigm of handling $\mathcal{G}_{n,m}$



Properties of $\mathcal{G}_{n,p}$

- Homogeneous in degree and dense when p is constant
- Impractical: typical real networks are heterogeneous&sparse

A tentative model for sparse graphs

When the graph has constant average degree

Consider a social network with average degree 150 (Dunbar's $\#$).
Let $p = \frac{150}{n}$. Does it work?

Too concentrated in degree

$D_i \sim \mathrm{Bin}(n-1, 150/n) \approx \mathrm{Poi}(150)$.
Chernoff $+$ Union bound implies concentration around 150.
e.g. $\Pr(D_i \leq 25) \leq 25\frac{e^{-150}150^{25}}{25!} \approx 25 \times 10^{-36} < 10^{-34}$.

Random graphs with a given degree sequence

Degree sequence of an n-vertex graph G

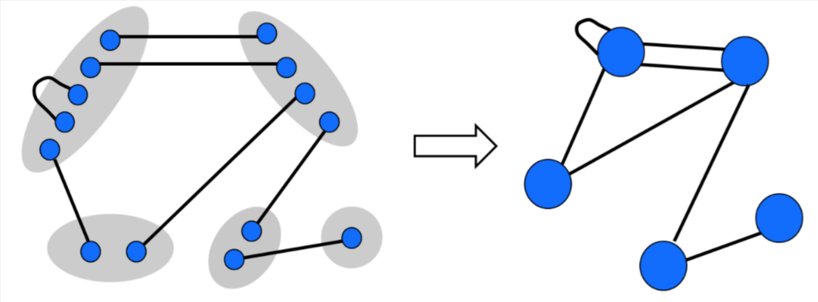
$n_0, n_1, ...n_n$ are integers.
n_i = number of vertices in G with degree exactly i.
$\sum n_i = n, \sum i * n_i = 2m$

Random graphs with specified degree sequence

Introduced by Bela Bollobas around 1980.
Produced by a random process:
Step 1. Decide what degree each vertex will have.
Step 2. Blow each vertex up into a group of 'mini-vertices'.
Step 3. Uniformly randomly, perfectly match these vertices.
Step 4. Merge each group into one vertex.
Finally, fix multiple edges and self-loops if you like

Example

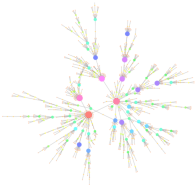
$n = 5, n_0 = 0, n_1 = 1, n_2 = 2, n_3 = 0, n_4 = 1, n_5 = 1$



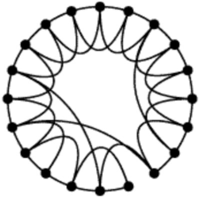
Other random graph models

Practical graphs are formed organically by “randomish” processes.

Preferential attachment model
Propsed by Barabasi&Albert in 1999
Scale-free network
First by Scottish statistician Udny Yule
in 1925 to study plant evolution



Rewired ring model
Propsed by Watts&Strogatz in 1998
Small world network



Threshold phenomena

Threshold: the most striking phenomenon of random graphs.
Extensively studied in Erdös-Rényi model $\mathcal{G}_{n,p}$.

Threshold functions

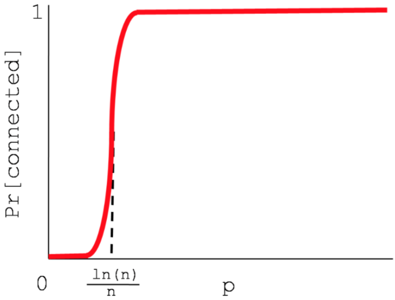
Given $f(n)$ and event E, if E does not happen on $\mathcal{G}_{n,o(f)}$ whp but happens on $\mathcal{G}_{n,w(f)}$ whp, $f(n)$ is a threshold function of E.

Sharp threshold functions

Given $f(n)$ and event E, if E does not happen on $\mathcal{G}_{n,cf}$ whp for any $c < 1$ but happens whp for any $c > 1$, $f(n)$ is a sharp threshold function of E.

Example

$f(n) = \frac{\ln n}{n}$ is a sharp threshold function for connectivity.



$f(n) = \frac{1}{n}$ is a sharp threshold function for giant component.

$f(n) = \frac{1}{n}$ is a threshold function for cycles.

Application: Hamiltonian cycles in random graphs

Objective

Find a Hamiltonian cycle if it exists in a given graph.
NP-complete, but ...
Efficiently solvable w.h.p. for $\mathcal{G}_{n,p}$, when p is big enough.

How?

A simple algorithm (use adjacency list model):

- Initialize the path to be a vertex.
- repeatedly use an unused edge to extend or rotate the path until a Hamiltonian cycle is obtained or a failure is reached.

Performance

Running time $\leq$ #edges $\Rightarrow$ inaccurate.
This does not matter if accurate w.h.p.
Challenge: hard to analyze, due to dependency.

A closer look at the algorithm

Essentially, extending or rotating is to sample a vertex.If an unseen vertex is sampled, add it to the path. When all vertices are seen, a Hamiltonian path is obtained, and almost end.

Familiar? Yes! Coupon collecting.
If we can modify the algorithm so that sampling at every step is uniformly random over all vertices, coupon collector problem results guarantee to find a Hamiltonian path in polynomial time. It is not so difficult to close the path.

Improvements

- Every step follows either unseen or seen edges, or reverse the path, with certain probability.
- Independent adjacency list (unused edges accessed by query), simplifying probabilistic analysis of random graphs

Modified Hamiltonian Cycle Algorithm

Under the independent adjacency list model

- Start with a randomly chosen vertex
- Repeat:
  - reverse the path with probability $\frac{1}{n}$
  - sample a used edge and rotate with probability $\frac{|used\_edges|}{n}$
  - otherwise, sample an unused edge (and rotate when necessary)
- Until a Hamiltonian cycle is found or FAIL(no unused edges)

An important fact

Let V_t be the head of the path after the t-th step. If the unused_edges list of the head at time $t-1$ is non-empty, $\Pr(V_t = u_t | V_{t-1} = u_{t-1}, ...V_0 = u_0) = \frac{1}{n}$ for $\forall u_i$.

Coupon collector results apply: If no unused edges lists are exhausted, a Hamiltonian path is found in $O(n \ln n)$ iterations w.h.p., and likewise for closing the path.

Performance and Efficiency

Theorem

If in the independent adjacency list model, each edge (u, v) appear on u's list with probability $q \geq \frac{20 \ln n}{n}$, The algorithm finds a Hamiltonian cycle in $O(n \ln n)$ iterations with probability $1 - O(\frac{1}{n})$.

Basic idea of the proof

Fail $\Rightarrow$

- $\mathcal{E}_1$: no unused-edges list is exhausted in $3n \ln n$ steps but fail.
  - $\mathcal{E}_{1a}$: Fail to find a Hamiltonian path in $2n \ln n$ steps.
  - $\mathcal{E}_{1b}$: The Hamiltonian path does not get closed in $n \ln n$ steps.
- $\mathcal{E}_2$: an unused-edges list is exhausted in $3n \ln n$ steps.
  - $\mathcal{E}_{2a}$: $\geq 9 \ln n$ unused edges of a vertex are removed in $3n \ln n$ steps.
  - $\mathcal{E}_{2b}$: a vertex initially has $< 10 \ln n$ unused edges.

Proof: $\mathcal{E}_{1a}$ and $\mathcal{E}_{1b}$ have low probability

$\mathcal{E}_{1a}$: Fail to find a Hamiltonian path in $2n \ln n$ steps

The probability that a specific vertex is not reached in $2n \ln n$ steps is $(1 - 1/n)^{2n \ln n} \leq e^{-2 \ln n} = n^{-2}$.
By the union bound, $\Pr(\mathcal{E}_{1a}) \leq n^{-1}$.

$\mathcal{E}_{1b}$: The Hamiltonian path does not get closed in $n \ln n$ steps

$\Pr(\text{close the path at a specific step}) = n^{-1}$.
$\Rightarrow \Pr(\mathcal{E}_{1b}) = (1 - 1/n)^{n \ln n} \leq e^{-\ln n} = n^{-1}$.

Proof: $\mathcal{E}_{2a}$ and $\mathcal{E}_{2b}$ have low probability

$\mathcal{E}_{2a}$: $\geq 9 \ln n$ unused edges of a vertex are removed in $3n \ln n$ steps

The number of edges removed from a vertex v's unused edges list $\leq$ the number X of times that v is the head.
$X \sim \mathrm{Bin}(3n \ln n, n^{-1}) \Rightarrow \Pr(X \geq 9 \ln n) \leq (e^2/27)^{3 \ln n} \leq n^{-2}$.
By the union bound, $\Pr(\mathcal{E}_{2a}) \leq n^{-1}$.

$\mathcal{E}_{2b}$: a vertex initially has $< 10 \ln n$ unused edges

Let Y be the number of initial unused edges of a specific vertex.
$\mathbb{E}[Y] \geq (n-1)q \geq 20(n-1)\ln n/n \geq 19 \ln n$ asymptotically.
Chernoff bounds $\Rightarrow \Pr(Y \leq 10 \ln n) \leq e^{-19(9/19)^2 \ln n/2} \leq n^{-2}$.
Union bound $\Rightarrow \Pr(\mathcal{E}_{2b}) \leq n^{-1}$.

Altogether

$\Pr(fail) \leq \Pr(\mathcal{E}_{1a}) + \Pr(\mathcal{E}_{1b}) + \Pr(\mathcal{E}_{2a}) + \Pr(\mathcal{E}_{2b}) \leq \frac{4}{n}$.

The algorithm on random graph $\mathcal{G}_{n,p}$

Corollary

The modified algorithm finds a Hamiltonian cycle on random graph $\mathcal{G}_{n,p}$ with probability $1 - O(\frac{1}{n})$ if $p \geq 40\frac{\ln n}{n}$.

Proof

Define $q \in [0, 1]$ be such that $p = 2q - q^2$.
We have two facts:

- The independent adjacency list model with parameter q is equivalent to $\mathcal{G}_{n,p}$.
- $q \geq \frac{p}{2} \geq 20\frac{\ln n}{n}$.