Probabilistic Method and Random Graphs

Lecture 7. Random Graphs [1]

Xingwu Liu

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

[1] The slides are mainly based on Lecture 13 of Ryan O'Donnell's lecture notes of *Probability and Computing* and Chapter 5 of the textbook *Probability and Computing*.

Questions, comments, or suggestions?

A recap of Lecture 6

Poisson approximation theorem

- $(X_1^{(m)}, X_2^{(m)}, ...X_n^{(m)}) \sim (Y_1^{(\mu)}, Y_2^{(\mu)}, ...Y_n^{(\mu)} | \sum Y_i^{(\mu)} = m)$
- $\mathbb{E}[f(X_1^{(m)}, ...X_n^{(m)})] \leq e\sqrt{m}\mathbb{E}[f(Y_1^{(m)}, ...Y_n^{(m)})]$
  - $Pr[\mathcal{E}(X_1^{(m)}, ...X_n^{(m)})] \leq e\sqrt{m}Pr[\mathcal{E}(Y_1^{(m)}, ...Y_n^{(m)})]$
  - $e\sqrt{m}$ can be improved to 2, if f is monotonic in m

Application

- Max. load: $L(n, n) > \frac{\ln n}{\ln \ln n}$ with high probability
- Hashing
  - Hash table: accurate, time-efficient, space-inefficient
  - Info. fingerprint: small error, time-inefficient, space-efficient
  - Bloom filter: small error, time-efficient, more space-efficient

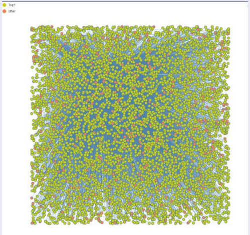
Motivation of studying random graphs

Gigantic graphs are ubiquitous

- Web link network: Teras of vertices and edges
- Phone network: Billions of vertices and edges
- Facebook user network: Billions of vertices and edges
- Human neural networks: 86 Billion vertices, $10^{14} - 10^{15}$ edges
- Network of Twitter users, wiki pages ...: size up to milllions

What do they look like?

- Impossible to draw and look
- What's meant by 'look like'?



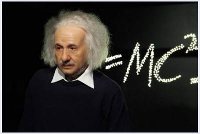
Looking through statistical lens

Part of the statistics

- How dense are the edges, $m = O(n)$ or $\Theta(n^2)$?
- Is it connected?
  - If not connected, the distribution of component size
  - If connected, diameter
- What's the degree distribution?
- What's the girth? How many triangles are there?

Feasible for a single graph?

Yes, but not of the style of a **scientist**



Scientists' concerns

Interconnection

- Do the features necessarily or just happen to appear?
- Do various gigantic graphs have common statistical features?
- What accounts for the statistical difference between them?

Prediction

- What will a newly created gigantic graph be like?
- How is one statistical feature, given some others?

Exploitation (algorithmical)

- How do the features help algorithms? Say, routing, marketing
- What properties of the graphs determine the performance?

Key to solution

Modelling gigantic graphs; random graphs are the best candidate

Definition of random graphs

Intuition: stochastic experiments

- God plays a dice, resulting in a random number
- God plays an amazing toy, resulting in a random graph
    - Amazing toy: a big dice with a graph on each facet

Axiomatic definition of random graphs

Random graph with n vertices

- Sample space: all graphs on n vertices
- Events: every subset of the sample space is an event
- Probability function: any normalized non-negative function on the sample space

An example

$\mathcal{G}_n$: uniform random graph on n vertices

The probability function has equal value on all graphs

Simple questions on $\mathcal{G}_n$

Random variable $X : G \mapsto$ the number of edges of G

- What's $\mathbb{E}[X]$?
- What's $Var[X]$?

Tough? Not easy, at least.
Big shots appeared!

A generative model of random graphs

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

Stochastic process:

input: n and $p \in [0, 1]$
output: indicators E_{ij}
for $i = 1 \cdot \cdot n$
for $j = i + 1 \cdot \cdot n$
$E_{ij} \leftarrow$ Bernoulli(p)

Proposed in 1959 by Gilbert (1923-2013, American coding theorist and mathematician). Motivated by phone networks.

In one word

$\mathcal{G}_{n,p}$ is an n-vertex graph the existence of each of whose edges is independently determined by tossing a p-coin.

Erdös&Rényi get the naming credit due to extensive work

An example: $p = \frac{1}{2}$

Uniform distribution over n-vertex graphs

$\mathcal{G}_{n,\frac{1}{2}} \sim \mathcal{G}_n$, the axiomatic definition
What does it look like?

The number of edges

In $\mathcal{G}_{n,\frac{1}{2}}$, the number of edges has $Bin\left(\binom{n}{2}, \frac{1}{2}\right)$ distribution.
Expectation: $\frac{n(n-1)}{4}$.
Variance: $\frac{n(n-1)}{8}$.
The expected degree of vertex i: $\frac{n-1}{2}$

Homogeneous degree distribution

Concentration theorem

In $\mathcal{G}_{n+1,\frac{1}{2}}$, all vertices have degree between $\frac{n}{2} - \sqrt{n \ln n}$ and $\frac{n}{2} + \sqrt{n \ln n}$ w.h.p.

Proof: Chernoff bound + Union Bound

Let D_i be the degree of vertex i.
$\Pr[D_i > \frac{n}{2} + \sqrt{n \ln n}] \leq e^{-(2\sqrt{\ln n})^2/2} = n^{-2}$.
Likewise, $\Pr[D_i < \frac{n}{2} - \sqrt{n \ln n}] \leq n^{-2}$.
By union bound, $\Pr[\frac{n}{2} - \sqrt{n \ln n} \leq D_i \leq \frac{n}{2} - \sqrt{n \ln n} \text{ for all } i] \geq 1 - \frac{2(n+1)}{n^2} = 1 - O(\frac{1}{n})$

Another generative model of random graphs

$\mathcal{G}_{n,m}$

Randomly *independently* assign m edges among n vertices.
Equiv: All n-vertex m-edge graphs, uniformly distributed.

Proposed by Erdös&Rényi in 1959, and
independently by Austin, Fagen, Penney and Riordan in 1959.
Hard to study, due to dependency among edges.
Can we decouple the edges? Yes, sort of.

Decoupling the edges

$\mathcal{G}_{n,m} \sim \mathcal{G}_{n,p} | (m \text{ edges exist})$
Recall the Poisson Approximation Theorem

Both are called Erdös-Rényi model.
$\mathcal{G}_{n,p}$ is more popular.

Application of the decoupling

Probability of having isolated vertices

In random graph $\mathcal{G}_{n,m}$ with $m = \frac{n \ln n + cn}{2}$, the probability that there is an isolated vertex converges to $1 - e^{-e^{-c}}$.

Proof (By myself)

Basically, follow the proof of the theorem about coupon collecting. It is reduced to $\mathcal{G}_{n,p}$ with $p = \frac{\ln n + c}{n}$.

Problem reduction

In $\mathcal{G}_{n,p}$ with $p = \frac{\ln n + c}{n}$, the probability that there is an isolated vertex converges to $1 - e^{-e^{-c}}$.

Proof

E_i: the event that vertex v_i is isolated in $\mathcal{G}_{n,p}$.
E: the event that at least one vertex is isolated in $\mathcal{G}_{n,p}$.

$$\Pr(E) = \Pr(\cup_{i=1}^{n} E_i)$$
$$= -\sum_{k=1}^{n}(-1)^k \sum_{1 \le i_1 < i_2 < \ldots < i_k \le n} \Pr(\cap_{j=1}^{k} E_{i_j}).$$

By Bonferroni inequalities,
$\Pr(E) \le -\sum_{k=1}^{l}(-1)^k \sum_{1 \le i_1 < \ldots < i_k \le n} \Pr(\cap_{j=1}^{k} E_{i_j})$, for odd l.

$$\Pr(\cap_{j=1}^{k} E_{i_j}) = (1-p)^{(n-k)k+\frac{k(k-1)}{2}} = (1-p)^{nk-\frac{k(k+1)}{2}}.$$
$$\Pr(E) \le -\sum_{k=1}^{l}(-1)^k \binom{n}{k} (1-p)^{nk-\frac{k(k+1)}{2}}, \text{for odd } l$$

$$\binom{n}{k}(1-p)^{nk-\frac{k(k+1)}{2}} > \frac{(n-k)^k}{k!}(1-p)^{nk-\frac{k(k+1)}{2}} \overset{n\to\infty}{=} \frac{e^{-ck}}{k!}.$$
$$\binom{n}{k}(1-p)^{nk-\frac{k(k+1)}{2}} < \frac{n^k}{k!}(1-p)^{nk-\frac{k(k+1)}{2}} \overset{n\to\infty}{=} \frac{e^{-ck}}{k!}$$

Continued proof

For odd l

$$\overline{\lim}_{n\to\infty} \Pr(E) \leq -\sum_{k=1}^{l} \frac{(-e^{-c})^k}{k!} = 1 - \sum_{k=0}^{l} \frac{(-e^{-c})^k}{k!}$$

For even l, likewise

$$\underline{\lim}_{n\to\infty} \Pr(E) \geq -\sum_{k=1}^{l} \frac{(-e^{-c})^k}{k!} = 1 - \sum_{k=0}^{l} \frac{(-e^{-c})^k}{k!}$$

Altogether

Let l go to infinity. We have
$\underline{\lim}_{n\to\infty} \Pr(E) = \overline{\lim}_{n\to\infty} \Pr(E) = 1 - e^{-e^{-c}}$.
So, $\lim_{n\to\infty} \Pr(E) = 1 - e^{-e^{-c}}$

Reflection on $\mathcal{G}_{n,p}$

Homogeneity in degree

Degree of each vertex is $\text{Bin}(n-1, p)$.
Highly concentrated, as proven

Dense for constant p

$m = \Theta(n^2)$ whp.
Billions of vertices with zeta edges, too dense

Unfit for real-world networks

Heterogeneous in degree distribution.
Sort of sparse

Remark

$\mathcal{G}_{n,p}$-type randomness does appear in big graphs.
Szemerédi Regularity Lemma (1975-1978)

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

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

Random graphs with a given degree sequence

Degree sequence of an n-vertex graph G

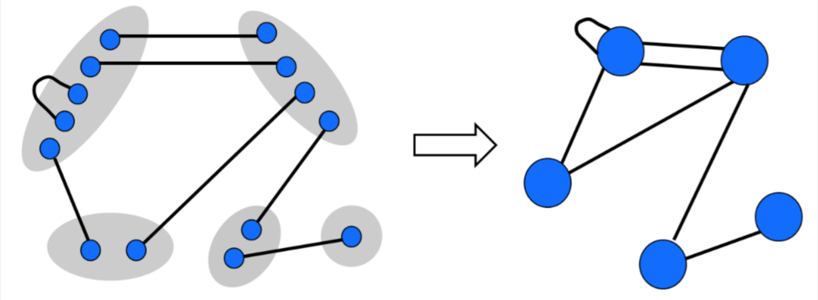
$n_0, n_1, ... n_n$ are integers.
$n_i =$ number of vertices in G with degree exactly i.
$\sum n_i = n, \sum i * n_i = 2m$

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

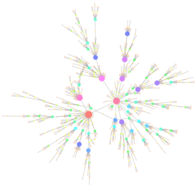
$n = 5, n_0 = 0, n_1 = 1, n_2 = 2, n_3 = 0, n_4 = 1, n_5 = 1$



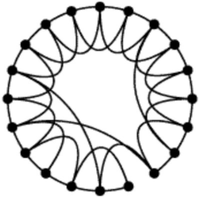
Other random graph models

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 the Erdös-Rényi model $\mathcal{G}_{n,p}$.

Threshold functions

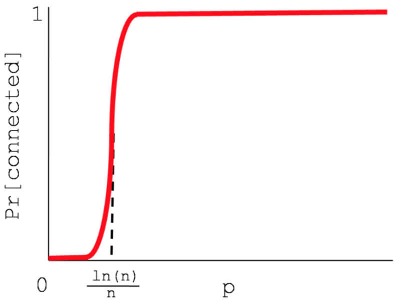
Given $f(n)$ and event E, if E does not happen on $\mathcal{G}_{n,o(f)}$ whp but happens on $\mathcal{G}_{n,w(f)}$ whp, $f(n)$ is a threshold function of E.

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 large components.

$f(n) = \frac{1}{n}$ 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, simplifying probabilistic analysis of random graphs (for general purpose)

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 $\frac{|used-edges|}{n}$
  - select the first unused edge with the rest probability
- Until a Hamiltonian cycle is found or fail

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.
$X \sim Bin(3n \ln n, n^{-1}) \Rightarrow \Pr(X \geq 9 \ln n) \leq (e^2/27)^{3 \ln n} \leq n^{-2}$.
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 $\mathcal{G}_{n,p}$.
- $q \geq \frac{p}{2} \geq 20\frac{\ln n}{n}$.