Graph Isomorphism course, Spring 2017
Instructor: László Babai
Notes by Angela Wu and the instructor
Thursday, April 20, 2017

8 Day 8, ThWk4

8.1 Erdős-Rényi model of random graphs

Last time we talked about how Naive Refinement can solve Graph Isomorphism for almost all graphs. We define what "almost all" means.

Definition 8.1 (Erdős-Rényi random graph)**.** Denote by $\mathcal{G}_{n,p}$ the probability distribution over the $2^{\binom{n}{2}}$ graphs on a given set V of n vertices defined below. The **Erdős-Rényi random graph** is a graph chosen according to $\mathcal{G}_{n,p}$.

A *Bernoulli trial* with probability p of success is a ranomd variable that takes value 1 with probability p ("success") and value 0 with probability $1-p$ ("failure").

Independently for each of the $\binom{n}{2}$ pairs $\{u, v\}$ of vertices, perform a Bernoulli trial with probability p of success, and make u and v adjacent if the trial is succeeds; non-adjacent if it fails.

DO 8.2. For a graph X chosen from the distribution $\mathcal{G}_{n,p}$ (notation: $X \sim \mathcal{G}_{n,p}$), the expected number of edges is $p\binom{n}{2}$ and the variance of the number of edges is $p(1-p)\binom{n}{2}$. Compute the expected number of triangles in X and the variance $V(p,n)$ of the number of triangles; asymptotically evaluate the latter when p is fixed and $n \to \infty$. Your answer to this last question should be of the form $V(p,n) \sim an^b$ where a, b are constants – determine a and b. The "asymptotic equality" $a_n \sim b_n$ of the sequences a_n, b_n means $\lim_{n\to\infty} a_n/b_n = 1$.

Definition 8.3 ("With high probability")**.** Let A_n be a sequence of events in a sequence of probability spaces, for $n \in \mathbb{N}$. A sequence A_n of events happens **with high probability (w.h.p.)** if $\mathbb{P}[A_n] \to 1$ as $n \to \infty$. We say that an event A_n happens **with very high probability (w.v.h.p.)** if there exists $0 < c < 1$ such that $\mathbb{P}[A_n] > 1 - c^n$.

The following three results are from Babai–Erdős–Selkow. We consider the uniform Erdős–Rényi graphs $\mathcal{G}_{n,1/2}$.

Lemma 8.4. *There exists a constant $\epsilon > 0$ such that with high probability the top n^ϵ vertex degrees are distinct.*

Let $S = (s_1, \ldots, s_k)$ be a list (ordered set) of vertices and let $\widetilde{S} = \{s_1, \ldots, s_k\}$. Let $x \in V \setminus \widetilde{S}$. Define by $\text{code}(x)$ to be the string in $\{0,1\}^{|S|}$ such that the i-th entry is the indicator for the adjacency $\{x, s_i\}$ (1 if they are adjacent, 0 otherwise).

Lemma 8.5. *Let S be the list of vertices of highest $3\log_2 n$ degrees. Then, w.h.p. all codes $\text{code}(x)$ ($x \in V \setminus \widetilde{S}$) are distinct.*

The proof in BES yields the bound $1/n^{1/7}$ on the probability that not all codes are distinct.

Corollary 8.6 (BES)**.** *With high probability,* Naive Refinement *completely splits a random graph in 2 rounds, and thereby solves GI for almost all graphs against any graph in linear time.*

Theorem 8.7 (Babai–Kučera (1979))**.** *Consider $\mathcal{G}_{n,1/2}$. W.v.h.p., a random graph is completely split by* Naive Refinement *in* 3 *rounds.*

The following lemma is the first step in the proof of the result.

HW 8.8. W.v.h.p. the random graph has $\Omega(\sqrt{n})$ distinct degrees. In fact, the probability that this fails is $< n^{-cn}$ for some constant $c > 0$.

DO 8.9 ($\sim$1973)**.** GI is Cook-equivalent to "Orbits of $\mathrm{Aut}(X)$," the decision problem described by "given $x, y \in V$, does there exist $\alpha \in \mathrm{Aut}(X)$ such that $x^\alpha = y$?"
Hint: Solve for vertex-colored graphs.

DO 8.10 (Babai–Mathon (1978))**.** GI is Cook-equivalent to both (1) computing $|\mathrm{Aut}(X)|$, and (2) finding a set of generators of $\mathrm{Aut}(X)$.

8.2 "Tower of groups" method

The method appears in a 1979 paper by Babai. The main result of that paper is that GI for vertex-colored graphs of bounded color multiplicity can be tested in Las Vegas polynomial time (see definition below). The algorithm was subsequently derandomized by Furst–Hopcroft–Luks (1980). These results are explained in these notes.

Definition 8.11 (Monte-Carlo algorithm)**.** A **Monte-Carlo algorithm** is a randomized algorithm of which the success probability is at least $1 - \epsilon$, where $\epsilon > 0$ is set by the user. The cost of the algorithm is proportional to $\log(1/\epsilon)$.

Definition 8.12 (Las-Vegas algorithm)**.** A **Las Vegas algorithm** is an algorithm that never errs, but probability $\leq \epsilon$ reports failure, where $\epsilon > 0$ is set by the user. The cost of the algorithm is proportional to $\log(1/\epsilon)$.

Definition 8.13 (Random element)**.** When speaking of a "random element" of a non-empty finite set S, we mean an element from the uniform distribution over S, unless expressly specified otherwise.

Consider a chain of subgroups of a finite group G,

$$G = G_0 \geq G_1 \geq \cdots \geq G_m = 1.$$

We make the following assumptions on access to this chain of groups.

(0) Black-box access to G. Not very precisely, this means that all group elements have names (strings of equall length over a finite alphabet) and we have oracles that perform group operations (multiplication, inversion, recognizing the identity).

(1) Each G_i is recognizable in G: given $g \in G$ and $i \leq m$, an oracle determines whether $g \in G_i$.

(2) An upper bound M on the jumps is given: $(\forall i)(|G_{i-1} : G_i| \leq M)$.

(3) Independent random elements of G_0 are available.

(4) The order of G is given.

(5) A set of generators of G is given.

(An "oracle" is a device that accepts certain types of queries.)

Theorem 8.14 (Tower-of-groups, randomized (B 1979))**.** *Under assumptions (0), (1), (2), (3), there is a randomized algorithm that will, w.h.p., (a) find the order of each G_i, (b) find generators for each G_i, and (c) generate random elements of each G_i, at the a cost of* $\text{poly}(m, M)$ *group operations and membership queries. If additionally we assume (4) then the algorithm is Las Vegas.*

This result was subsequently derandomized by Furst, Hopcroft, and Luks (1980).

Theorem 8.15 (Tower-of-groups, deterministic (FHL 1980))**.** *Under assumptions (0), (1), (2), (5), there is a deterministic algorithm that will, w.h.p., (a) find the order of each G_i, (b) find generators for each G_i, and (c) generate random elements of each G_i, at the a cost of* $\text{poly}(m, M)$ *group operations and membership queries.*

Let T_i be a set of right coset representatives of G_i in G_{i-1} A collection of the form $\mathcal{T} = (T_1, \ldots, T_m)$ is a **coset table** for this tower. To prove Theorem 8.14 and Theorem 8.15, it suffices to find a coset table for the subgroup chain (see DO exercises below).

DO 8.16 (Prove (a))**.** Show that $|G_i| = \prod_{j>i} |T_j|$.

DO 8.17 (Prove (b))**.** $G_i = \langle \bigcup_{j>i} T_j \rangle$.

DO 8.18. $G_{i-1} = G_i T_i$ uniquely (each $g \in G_{i-1}$ can uniquely be written as $g = ht$ where $h \in G_i$ and $t \in T_i$). Infer that $G_0 = T_m \cdot T_{m-1} \cdot \ldots \cdot T_1$ uniquely.

DO 8.19 (Prove (c))**.** To obtain a random element of G, take a product of the form $t_m t_{m-1} \cdots t_1$ where t_i is a random element of T_i.

Definition 8.20. We say that $\mathcal{T} = (T_1, \ldots, T_m)$ is a **partial coset table** if for every $i \le m$,

- $T_i \subseteq G_{i-1}$
- $1 \in T_i$
- no two elements of T_i are in the same right coset of G_i, i.e., if $x, y \in T_i$ and $xy^{-1} \in G_i$ then $x = y$.

The algorithm will start from the smallest partial coset table ($T_i = \{1\}$ for all i) and gradually build it up to a full coset table.

8.2.1 Sifting

First we present a subroutine, Sift, due to Schreier–Sims, that takes an element $g \in G$ and either represents it as a product $g \in T_m \cdots T_1$ from the current partial coset table $(T_1, \ldots, T_m)$, or uses g to add an element to the coset table.

Procedure Sift$(\mathcal{T}, g)$

Input: access (0) and (1) to the subgroup chain
partial coset table $(T_1, \ldots, T_m)$ and an element $g \in G$

Output: either a representation $g = t_m \cdots t_1$ where $t_i \in T_i$

or a new element to be added to the coset table.
Loop invariant: $g \in G_{i-1}$

for $i = 1$ **to** m
  for $t \in T_i$
    if $gt^{-1} \in G_i$
    then $t_i \leftarrow t$, $g \leftarrow gt^{-1}$ (: peeling off a coset rep :)
      exit inner "for" loop
      (: no more $t \in T_i$ will be tested, we move to $i \leftarrow i+1$:)
  end(for) (: $(\forall t \in T_i)(gt^{-1} \notin G_i)$:)
  add g to T_i
  return updated coset table
  exit SIFT
end(for) (: g "sifted all the way down" :)
return $(t_1, \dots, t_m)$ (: $g = t_m t_{m-1} \dots t_1$:)
end(Procedure)

DO 8.21. Prove the correctness of the procedure.

DO 8.22. Let $N = \sum_{i=1}^{m} |G_{i-1} : G_i|$. The cost of SIFT is $\leq N$ group operations and the same number of membership queries (membership in G_i).

8.2.2 Tower of Groups, randomized

This procedure appears in [B 1979].
Procedure TOWER-OF-GROUPS, RANDOMIZED
Input: access (0), (1), (2), (3) to the subgroup chain
Output: coset table $\mathcal{T} = (T_1, \dots, T_m)$
Loop invariant: $\mathcal{T}$ is a partial coset table

  Initialization
    for $i = 1 \dots m$
      $T_i = \{1\}$
  Body of algorithm
  repeat a sufficient number of times – this will be $Mm + r$, r determined below.
    pick a random $g \in G_0$
    SIFT($\mathcal{T}, g$)
  end(repeat)
  return $\mathcal{T} = (T_1, \dots, T_m)$
  end(Procedure)

DO 8.23. In applying SIFT to random elements of $g \in G$, some get stuck in T_j for some $j > i$, the others reach G_i (after having peeled off a sequence of coset representatives). Prove: those that reach G_i form a sequence of independent random elements of G_i.

Determining the number of iterations.

Proposition 8.24. *Let $\epsilon > 0$ and let $r > M(\ln(Mm) - \ln \epsilon)$. Then the probability that after $Mm+r$ rounds, the coset table is not full, is less than ϵ.*

Proof. During the $nM + r$ rounds, at most mM elements get stuck in the coset table; all the others sift all the way down, providing a shower of at least r independent random elements for each G_i. It follows that

$$\mathbb{P}[\text{a coset of } G_i \text{ in } G_{i-1} \text{ is missed}] \leq (1 - 1/M)^r < \mathrm{e}^{-r/M}.$$

So,

$$\mathbb{P}[\text{coset table is not full }] < Mm\mathrm{e}^{-r/M} \leq \epsilon$$

as long as $r > M(\ln(Mm) - \ln \epsilon)$. □

DO 8.25. The partial coset table $\mathcal{T} = (T_1, \ldots, T_m)$ is full if and only if $|G| + \prod_{i=1}^{m} |T_i|$.

DO 8.26. Suppose we add assumption (4): $|G_0|$ is known. Then, this algorithm is Las Vegas (honestly reports failure, which occurs with probability $< \epsilon$).

8.2.3 Tower of Groups, deterministic

Next we present the FHL derandomization (1980) of this procedure.
Procedure Tower-of-groups, deterministic
Input: access (0), (1), (5) to the subgroup chain: a list S of generators of G is given
Output: coset table $\mathcal{T} = (T_1, \ldots, T_m)$
Loop invariant: $\mathcal{T}$ is a partial coset table

```
Initialization
    for i = 1 ... m
        T_i = {1}
Body of algorithm
for s ∈ S
    Sift(T, s)
end(for)
for t, t' ∈ ⋃_{i=1}^m T_i
    Sift(T, tt')
end(for)
return T = (T_1, ..., T_m)
end(Procedure)
```

HW 8.27. Prove correctness of the procedure (i.e., prove that in the end, the coset table is full).

DO 8.28. The number of rounds (siftings) is $\leq |S| + (mM)^2$.

Remark 8.29. A more efficient termination rule was found more than a decade earlier by C. C. Sims. His algorithm was analyzed by Knuth (1982–91).

8.3 Graphs with bounded color multiplicity

A vertex-colored graph is a triple $X = (V, E, f)$, where $f : V \to \{\text{colors}\}$ is the coloring. We write $C_i = f^{-1}(i)$ for the i-th color class. The *multiplicity* of color i is $|C_i|$.

Theorem 8.30. GI *of graphs with bounded color multiplicity can be tested*

(i) [B 1979] in Las Vegas (defined below) polynomial time

(ii) [FHL 1980] in deterministic polynomial time

We will prove this by determining $\mathrm{Aut}(X)$ for a vertex-colored graph $X = (V, E, f)$ that has bounded color multiplicity.

First we set some notation: Let $X = (V, E, f)$. Name the color classes $V = C_1 \sqcup \cdots \sqcup C_m$. Let $n_i = |C_i|$, so $n = n_1 + \cdots + n_m$. The number of potential isomorphisms is $\prod n_i!$. Let d be a bound on the color classes, so $(\forall i)(|C_i| \le d)$.

Denote by E_{ij} the set of edges between C_i and C_j. Denote by E_{ii} the set of edges within C_i.

We will build a tower of groups.

Here we build the "beginning" of the tower. Let X_0 be the colored set $X_0 := (V, \emptyset, f)$. Then, $\mathrm{Aut}(X_0) = S_{n_1} \times \cdots \times S_{n_m}$. Let $X_1 = (V_1, E_{11}, f)$, let $X_2 = (V, E_{11} \cup E_{12}, f))$, etc. Then, $X_{\binom{k+1}{2}} = X$. Let $G_i = \mathrm{Aut}(X_i)$. Then we have:

$$G_0 \ge G_1 \ge \cdots \ge G_{\binom{k+1}{2}} = \mathrm{Aut}(X). \quad (1)$$

DO 8.31. Show that indeed $G_i \le G_{i-1}$.

HW 8.32. $|G_{i-1} : G_i| \le (d!)^2$.

But, is G_i recognizable? Yes, because Aut("anything") is recognizable. By "anything" we really mean any explicit mathematical object.

To complete the chain in Equation (1), we append a stabilizer chain of $\mathrm{Aut}(X)$, explained below.

Definition 8.33. Let $H \le S_n$. A **stabilizer chain** is formed by stabilizing one more point in $[n]$ at every step in the chain: let H_i be the pointwise stabilizer of the set $[i]$. So

$$H = H_0 \ge H_1 \ge \cdots \ge H_n = 1.$$

DO 8.34. $|H : H_x| \le n$ for every $H \le S_n$ and $x \in [n]$.

DO 8.35. Show that the members of the stabilizer chain of the automorphism group of a graph are recognizable.

DO 8.36. Let $G \le S_n$ be the automorphism group of a colored set with color multiplicity $\le d$ and let $H \le G$ and $x \in [n]$. Then $|H : H_x| \le d$.

DO 8.37. Complete the proof of Theorem 8.30.

We shall say that a permutation group is "given" or "known" if a list of generators is given/known.

Theorem 8.38 (FHL 1980)**.** *Given $G \le S_n$ and $\sigma \in S_n$, membership of σ in S_n can be determined in polynomial time; and the order of G can be found in polynomial time.*

DO 8.39. Prove this result. (Apply the Tower-of-Groups method to the stabilizer chain.)

Definition 8.40 (Normal closure)**.** Let $H \le G$. The **normal closure** of H in G is the smallest normal subgroup of G containing H, i.e., the group generated by all conjugates of H.

DO 8.41 (FHL, 1980)**.** Given $H \le G \le S_n$ we can find the normal closure of H in G in polynomial time.