Lecture 7:

Testing dense graphs :

Bipartiteness

Testing Properties of Dense Graphs

Previously - graphs sparse, degree bounded by d, adjacency list representation

Next lectures - adjacency matrix representation, dense graphs, no degree bounds

Adjacency Matrix Model

G represented by matrix $A = \begin{bmatrix} & & \\ & A_{ij} & \\ & & \end{bmatrix}$

s.t. can query A_{ij} in one step

$A_{ij} = 1$ if $(i,j) \in E$, 0 o.w.

Distance from property P

$P \leftarrow$ set of graphs closed under permutations (relabeling of node names)

<u>def</u> G is ε-far from P if must change $\geq \varepsilon n^2$ entries in A to turn G into a member of P

Testing "sparse" properties in this model:

<u>all</u> graphs are ε-close to connected so trivial tester says "PASS" on all inputs

Bipartiteness

equivalent definitions

- can color nodes red / blue s.t. no edge monochromatic ↑
- can partition nodes into (V_1, V_2) s.t.

$$\underbrace{\nexists \underset{(u,v)}{\overset{\shortparallel}{e}} \in E \quad \text{s.t.} \quad \left.\begin{array}{l} u,v \in V_1 \\ \text{or } u,v \in V_2 \end{array}\right\} \text{"violating" edge}}$$

i.e. not bipartite $\iff$ $\forall$ partitions $V = (V_1, V_2)$ $\exists$ violating edge

def. ε-far from bipartite

equivalent

- must remove $> \varepsilon n^2$ edges to make bipartite
- $\forall$ partitions $V = (V_1, V_2)$, $> \varepsilon n^2$ violating edges

Testing Algorithms

- testing exact bipartiteness (not sublinear)
  BFS (linear time)

- Proposed property testing algorithm:

runtime independent of n

Passes bipartite graphs but does it fail far-from-bipartite graphs?

Pick sample of nodes of size $\Theta\left(\frac{1}{\varepsilon^2} \log \frac{1}{\varepsilon}\right)$
Consider induced subgraph on sample
If bipartite, output "pass"
else output "Fail"

This actually works!

A first attempt

Consider G, ε-far from bipartite

$\forall$ partitions (V_1, V_2) have $\geq \varepsilon n^2$ violating edges

$\Rightarrow$ $\forall (V_1, V_2)$ sample of size $m = \Theta(\frac{1}{\varepsilon} \log \frac{1}{\delta})$ will hit (V_1, V_2)-viol-edge with prob

$\geq 1 - (1-\varepsilon)^{\frac{c}{\varepsilon}\log\frac{1}{\delta}} = 1 - e^{-c\log 1/\delta} \geq 1-\delta$

for good choice of c

so what is the problem?

- lets not worry about time, just query complexity
- but how do we know that e violates (V_1, V_2)? & just because it violates (V_1, V_2) doesn't mean it violates all partitions!
- should we try all 2^n partitions?

Algorithm 0 [horrible runtime, but maybe query complexity ok?]

Pick $m = \Theta(?)$ random edgeslots & query

$\forall$ partitions (V_1, V_2)

violating$_{V_1, V_2}$ $\leftarrow$ # violating edges in sample wrt V_1, V_2

If all violating$_{V_1, V_2} > 0$ output FAIL
else PASS

How many queries needed?

- bipartite always passes

- if G is ε-far

$\Rightarrow \forall V_1, V_2 \quad \exists \geq \varepsilon n^2$ violating edges

$\Rightarrow \forall V_1, V_2 \quad \Pr[\text{see violating edge for } V_1, V_2] \geq 1 - \delta$

depends on # samples

$\Rightarrow \Pr[\forall V_1, V_2 \text{ see viol edge for } V_1, V_2] \geq \underbrace{1 - 2^n \cdot \delta}_{\text{union bound}}$

so need $\delta < \frac{1}{2^n}$?

this would require $m = \Theta(\frac{1}{\varepsilon} \log \frac{1}{\delta})$

$\approx \underbrace{\Theta(\frac{n}{\varepsilon})}_{\text{sublinear in } n^2, \text{ but want better!}}$

<u>Problem</u> do we really need a union bound?

<u>or</u> do we really need to try <u>all</u> partitions?

many have similar #'s of violating edges, few "representatives"

can we just pick a few partitions that are close to <u>all</u> partitions?

Algorithm 1

1. Pick U, U' randomly from V

$U = \Theta(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$ nodes — used to define a set of partitions

$U' = \Theta(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon})$ nodes — pair off + think of as edges

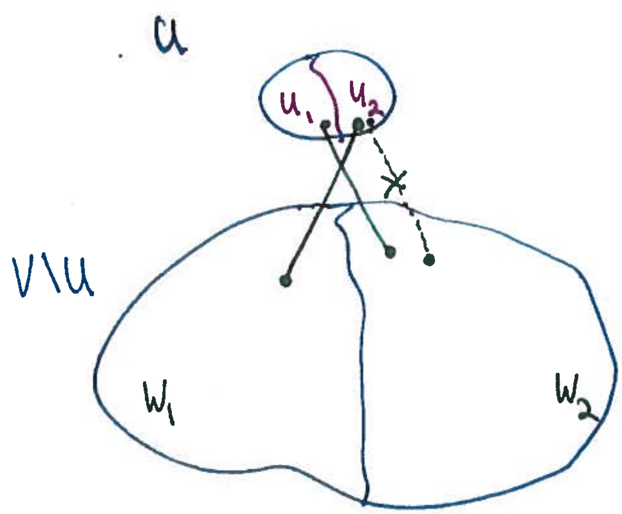
$U' = \{u_1, v_1, u_2, v_2 \dots\}$

$P = \{(u_1,v_1)(u_2,v_2)\dots\}$ pairs

If U not bipartite, FAIL

2. $\forall$ partitions of U into U_1, U_2
   - induce partition on rest of graph

Consider only $2^{|U|}$ partitions, is this enough?



if U is not bipartite, FAIL!

Partition: $\forall v \in V$ (including $v \in U$)
- if v has nbr in U_2, put in W_1
- if " " " " U_1 " " W_2
- " " " " " both $\Rightarrow$ bad partition (continue to next partition)
- " " " " " neither, put in W_1

$\forall v$ this can be computed in $O(|U|)$ time!

Don't need to compute for all $v \in V$, just for all $v \in U'$

- Count how many $(u,v) \in P$ violate W_1, W_2

Pass if fraction $\le \frac{3}{4} \cdot \varepsilon$

o.w. Continue to next partition

Why pass if any violation? because we aren't checking all W_1, W_2

3. Fail

Analysis

- if G bipartite:

  not immediate that it passes!

  the "right" partition might not be one that we try!

let $V = (Y_1, Y_2)$ be bipartite partition (no violating edges)

For sample U,

$U_1 \leftarrow Y_1 \cap U$

$U_2 \leftarrow Y_2 \cap U$

(note: U_1, U_2 is partition of U)

Now, use U_1, U_2 to partition V as in step 2: $W_1^{U_1,U_2}\ W_2^{U_1,U_2}$

Main Question: How close is $W_1^{U_1,U_2}\ W_2^{U_1,U_2}$ to Y_1, Y_2? ← how many extra violating edges can it have?

how can it differ?

only for v <u>without nbr in U</u>

note, if v has edge to <u>both</u> W_1, W_2 then contradicts that $Y_1\ Y_2$ is a bipartition

- v with small degree $(< \frac{\varepsilon}{4} n) = A$
- v with high degree $(\geq \frac{\varepsilon}{4} n) = B$

violating edges in $W_1^{U_1,U_2}\ W_2^{U_1,U_2}$:

$$\le 0 + \frac{\varepsilon}{4} n \cdot n + n \cdot \square$$

0: # violating edges of Y_1, Y_2

$\frac{\varepsilon}{4} n$: max degree of $v \in A$

n: $|A|$

n: max degree of $v \in B$

$\square$: $|B|$

*1

Lemma $\Pr_{\text{choice of } U}\left[\le \frac{\varepsilon}{4} n \text{ high degree nodes in } V \text{ with no nbr in } U \right] \ge 7/8$

Pf

$\forall v$ of degree $\ge \frac{\varepsilon}{4} n$ $\quad \sigma_v = \begin{cases} 1 & \text{if } U \text{ has no nbr of } v \\ 0 & \text{o.w} \end{cases}$

$\forall$ other v, $\sigma_v = 0$

for high degree v: $E[\sigma_v] = \Pr[\sigma_v = 1]$

$= \left(\Pr[i^{th} \text{ node of } U \text{ isn't nbr of } v]\right)^{|U|}$

$\le \left(1 - \frac{\varepsilon}{4}\right)^{|U|} = \left(1 - \frac{\varepsilon}{4}\right)^{\frac{4}{\varepsilon}\log 32/\varepsilon} \le \frac{\varepsilon}{32}$

↑ since v is high degree

for low degree v: $E[\sigma_v] = 0$

$E\left[\sum_{v \in V} \sigma_v\right] \le \frac{\varepsilon}{32} n$

$\Pr\left[\sum \sigma_v \ge \underbrace{8 \cdot \frac{\varepsilon}{32} n}_{\frac{\varepsilon n}{4}}\right] \le \frac{1}{8}$ by Markov's ≠ $\boxtimes$

so # violating edges in $W_1^{u_1 u_2} W_2^{u_1 u_2}$: (whp)

$\le \frac{\varepsilon}{4} n^2 + \underbrace{n \cdot \frac{\varepsilon n}{4}}_{\text{with prob } \ge 7/8 \text{ from lemma}}$

$\le \frac{\varepsilon n^2}{2}$

$\Rightarrow E[\text{fraction of } (u,v) \in P \text{ violating } W_1^{u_1 u_2} W_1^{u_1 u_2}] \le \frac{\varepsilon}{2}$

so $\Pr[$ " " " " ... $\ge \frac{3}{4}\varepsilon] \ll \frac{1}{8}$ *2

↑ use Chernoff + # samples to show this

So $\Pr[\text{output fail}]$

$$\le \underbrace{\Pr[\text{output fail} \mid \text{too many high degree nods}]}_{\le 1} \cdot \underbrace{\Pr[\text{too many high degree nodes}]}_{\le 1/8}$$

$$+ \underbrace{\Pr[\text{output fail} \mid \text{not too many high degree nodes}]}_{\le 1/8} \cdot \underbrace{\Pr[\text{not too many high degree nodes}]}_{\le 1}$$

$$\le \frac{1}{8} + \frac{1}{8} = \frac{1}{4}$$

- if G ε-far from bipartite:

<u>all</u> partitions Y_1, Y_2 have $\ge \varepsilon n^2$ violating edges

in particular so does $W_1^{u_1 u_2}\ W_2^{u_1 u_2}$ $\forall u_1, u_2$

$\Pr[\text{fraction of } (u,v) \in P \text{ violating } W_1^{u_1 u_2} W_2^{u_1 u_2} \le \frac{3}{4}\varepsilon n^2] << \frac{1}{8 \cdot 2^{|U|}}$

use Chernoff + # samples

$\Pr[\text{all partitions of } U \text{ have } \ge \frac{3}{4}\varepsilon n^2 \text{ violations}] \ge 1 - \frac{1}{8}$

$\therefore$ $\Pr[\text{output pass}] < 1/8$ ∎

Comments

1) Can improve runtime to $\text{poly}(1/\varepsilon)$
2) proposed testing algorithm actually works
3) in adjacency list model (sparse graphs), need $\Omega(\sqrt{n})$ queries

Other problems: Partition properties

Similar ideas work:

Use random sample to implement oracle ← actually several oracles

which tells you how to do a global partition ← so pick oracle giving best global result

idea for Max Cut

like greedy 2-approx for maxcut!

pick random sample S

for each partition of S, create oracle (S_1, S_2):

put $v \in V \setminus S$ on side $U_1^{S_1,S_2}$ if $e(v, S_2) \geq e(v, S_1)$

& side $U_2^{S_1,S_2}$ o.w.

then estimate # edges between $(S_1 \cup U_1^{S_1,S_2})$ & $(S_2 \cup U_2^{S_1,S_2})$

Output max value

Analysis is a bit more complicated...

More: Can ask "Given ρ_{ij}'s for $1 \leq i < j \leq k$ is there partition of V into k sets $S_1 \cdots S_k$ s.t. edge density bet S_i & S_j is ρ_{ij}?"