

Testing Properties of Dense Graphs

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

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

Adjacency Matrix Model

G represented by matrix $A = \begin{bmatrix} & & \\ & & \\ & & \end{bmatrix}$
 st. can query A_{ij} in one step

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

Distance from property P ← set of graphs closed under permutations (relabeling of node names)

def. G is ϵ -far from P
 if must change $\geq \epsilon n^2$ entries in A
 to turn G into a member of P

Testing "sparse" properties in this model:

all 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.

$\exists e \in E$ s.t. $u, v \in V_1$ or $u, v \in V_2$ } "violating" edge
 ("u,v")

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

def. ϵ -far from bipartite

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

Testing Algorithms

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

- Proposed property testing algorithm:

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

Pick sample of nodes of size $\frac{1}{\epsilon}$
 Consider induced subgraph on sample
 If bipartite, output "Pass"
 else output "Fail"

runtime independent of n

$$O\left(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon}\right)$$

This actually works!

A first attempt

Consider G , ϵ -far from bipartite

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

\Rightarrow \forall sample of size $m = \Theta\left(\frac{1}{\epsilon} \log \frac{1}{\delta}\right)$

will hit (V_1, V_2) -viol-edge with prob

$$\geq 1 - (1 - \epsilon)^{\frac{1}{\epsilon} \log \frac{1}{\delta}} = 1 - e^{-\log \frac{1}{\delta}} \geq 1 - \delta$$

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

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

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

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

↑
depends on # samples

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

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

$\approx \Theta\left(\frac{n}{\epsilon}\right)$

sublinear in n^2 ,
but want better!

Problem

do we really need a union bound?

or do we really need to try all partitions?

↑
many have similar #'s of violating edges,
can we just pick a few "representatives"
that are close to all partitions?

Algorithm 1

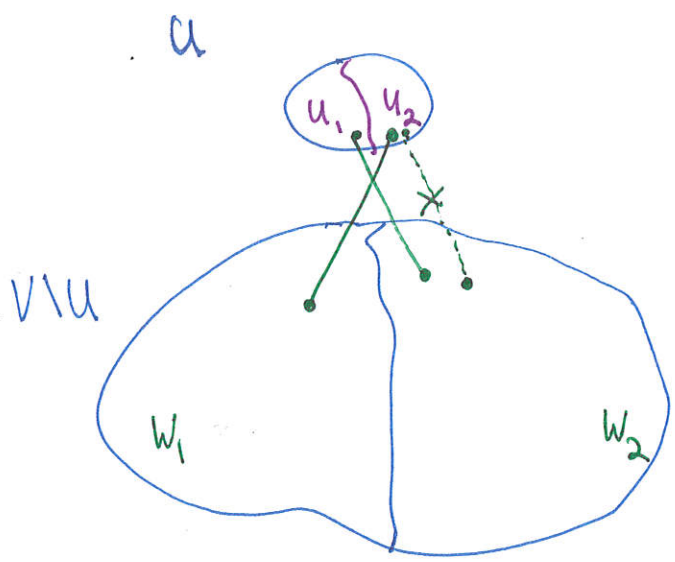
1. Pick u, u' randomly from V

$\theta(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$ nodes used to define a set of partitions
 $\theta(\frac{1}{\epsilon^2} \log \frac{1}{\epsilon})$ nodes \uparrow 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
 • " " " " " 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 $\leq \frac{3}{4} \cdot \epsilon$
 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 without nbr in U

note, if v has edge to both W_1, W_2 then contradicts that Y_1, Y_2 is a bipartition

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

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

$$\leq \underbrace{0}_{\substack{\uparrow \\ \text{\# violating edges of } Y_1, Y_2}} + \underbrace{\frac{\epsilon}{4} n \cdot n}_{\substack{\uparrow \quad \uparrow \\ \text{max degree of } v \in A \quad |A|}} + \underbrace{n \cdot \square}_{\substack{\uparrow \quad \uparrow \\ \text{max degree of } v \in B \quad |B|}}$$

*1

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

Pf

$\forall v$ of degree $\geq \frac{\epsilon}{4} n$ $\delta_v = \begin{cases} 1 & \text{if } U \text{ has no nbr of } v \\ 0 & \text{o.w} \end{cases}$
 $\forall \text{ other } v, \delta_v = 0$

for high degree v : $E[\delta_v] = \Pr[\delta_v = 1]$
 $= \left(\Pr[\text{ith node of } U \text{ isn't nbr of } v] \right)^{|U|}$
 $\leq \left(1 - \frac{\epsilon}{4} \right)^{|U|} = \left(1 - \frac{\epsilon}{4} \right)^{\frac{4}{\epsilon} \log^{32/\epsilon}} \leq \frac{\epsilon}{32}$

↑ since v is high degree

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

$$E\left[\sum_{v \in V} \delta_v\right] \leq \frac{\epsilon}{32} n$$

$$\Pr\left[\sum \delta_v \geq \underbrace{8 \cdot \frac{\epsilon}{32} n}_{\frac{\epsilon n}{4}}\right] \leq \frac{1}{8} \text{ by Markov's } \square$$

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

$$\leq \frac{\epsilon}{4} n^2 + \underbrace{n \cdot \frac{\epsilon n}{4}}_{\text{with prob } \geq 7/8 \text{ from lemma}}$$

$$\leq \frac{\epsilon n^2}{2}$$

$\Rightarrow E[\text{fraction of } (u,v) \in P \text{ violating } W_1^{u_1, u_2} W_2^{u_1, u_2}] \leq \frac{\epsilon}{2}$

so $\Pr[\text{fraction} \geq \frac{3\epsilon}{4}] \leq \frac{1}{8}$ *2
 ↑ use Chernoff + # samples to show this

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

$$\leq \Pr[\text{output fail} \mid \text{too many high degree nodes}] \cdot \Pr[\text{too many high degree nodes}]$$

$$+ \Pr[\text{output fail} \mid \text{not too many high degree nodes}] \cdot \Pr[\text{not too many high degree nodes}]$$

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

• if G ϵ -far from bipartite:

all partitions Y_1, Y_2 have $\geq \epsilon n^2$ violating edges

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

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

$\Pr[\text{all partitions of } U \text{ have } \geq \frac{3}{4} \epsilon n^2 \text{ violations}] \geq 1 - \frac{1}{8}$ ↑ use Chernoff + # samples

$\therefore \Pr[\text{output pass}] < \frac{1}{8}$ ▣

Comments

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

Other problems: Partition properties

Similar ideas work:

Use random sample to implement oracle \leftarrow actually several oracles
 which tells you how to do a global partition \leftarrow 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 p_{ij} 's for $1 \leq i < j \leq k$
 is there partition of V into k sets S_1, \dots, S_k
 s.t. edge density bet $S_i + S_j$ is p_{ij} ?"