Lecture 8

Testing dense graphs

- bipartiteness

Adjacency Matrix model

G represented by matrix A
s.t. can query A in
one step

$$A = \begin{bmatrix} & & \\ & A_{ij} & \\ & & \end{bmatrix}$$

1 if $(i,j) \in E$
0 o.w.

Distance from property P:

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

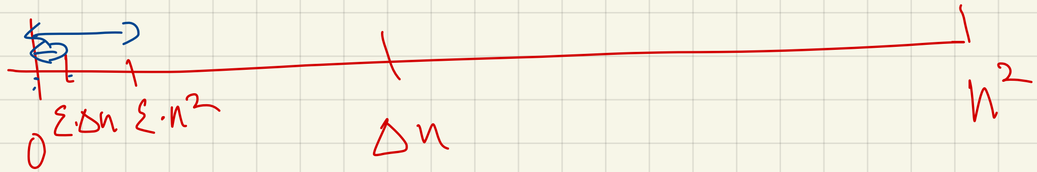
Testing "sparse" properties:

<u>all</u> graphs are ε-close to connected in this model

$\Rightarrow$ trivial tester outputs "PASS" w/o looking at graph

	Graph type	max degree	natural representation	notion of distance	
Previously	sparse	Δ	adjacency list	$\leq \varepsilon \cdot \Delta \cdot n$	edges changed
Now	dense	n	adjacency matrix	$\leq \varepsilon \cdot n^2$	" "

Should be easier to detect

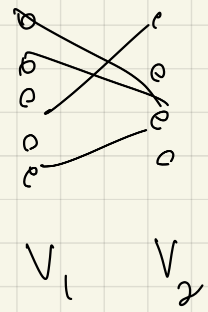


Bipartiteness:

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

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

equivalent definitions



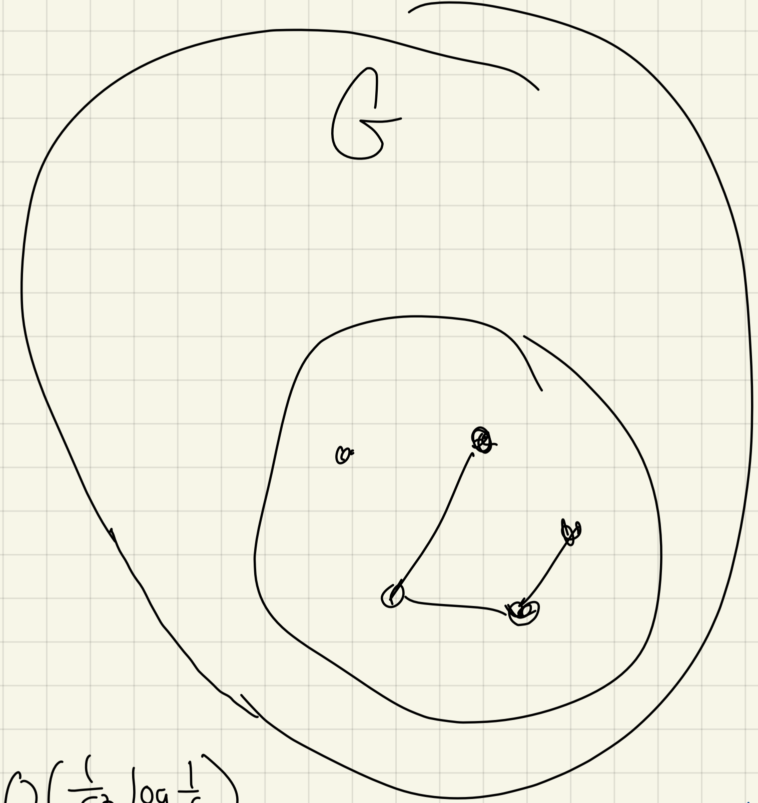
not bipartite $\Rightarrow$ $\forall (V_1, V_2)$ $\exists$ "violating edge"

ε-far from bipartite: (definition)

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

equivalent

Testing Algorithms:



- Testing exact bipartiteness:

  e.g. BFS

- Proposed testing algorithm:

  - Pick sample of nodes of size $O\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\right)$
  - Consider induced graph on sample ← ignore nodes not in sample; ignore edge s.t. ≥ 1 endpt is not in sample
  - If bipartite, output PASS ← e.g. BFS

    else output FAIL

This actually works!!

A first attempt at a proof?

if G bipartite, induced graph is bipartite, so algorithm passes ✓

if G ε-far from bipartite:

must remove εn^2 edges to make it bipartite

equivalently:

$\forall$ partition V_1, V_2 have $> \varepsilon n^2$ violating edges ($> \varepsilon$ fraction of slots in adj matrix)

$\Rightarrow$ $\forall (V_1, V_2)$ a sample of edges of size $\geq \Theta\left(\frac{1}{\varepsilon} \log \frac{1}{\delta}\right)$

hits a (V_1, V_2)-violating edge with prob $\geq 1 - (1-\varepsilon)^{\frac{c}{\varepsilon} \log \frac{1}{\delta}}$

$$\geq 1 - e^{-c \cdot \log \frac{1}{\delta}} = 1 - e^{-\log 1/\delta} = 1 - \delta$$

(set $c = 1$)

Great!?

need to hit violating edge for every partition

how is this an algorithm?

no edge violates all partitions

Lets try to use the "partition" defn of bipartiteness:

Algorithm 0

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

$\forall$ partitions (V_1, V_2):

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

If $\forall$ (V_1, V_2) violating$_{\sigma(V_1,V_2)} > 0$ then output FAIL

else output PASS

Wait! How small should δ be?

Recall: All partitions are bad

But: if any partition "looks" good, the algorithm outputs PASS

Probability any partition "looks" good:

for one partition (V_1, V_2), $\Pr[(V_1,V_2) \text{ looks good}] \le \delta$

for all partitions (V_1,V_2), $\Pr[\text{any } (V_1,V_2) \text{ looks good}] \le \underbrace{2^n \cdot \delta}_{\text{union bnd over } 2^n \text{ partitions}}$

$\therefore$ need $\delta \ll \frac{1}{2^n}$

would imply $\underset{\substack{\uparrow \\ \text{Sampled nodes}}}{m} = \Theta\left(\frac{1}{\varepsilon}\log\frac{1}{\delta}\right) = \Theta\left(\frac{1}{\varepsilon}\log\frac{1}{\frac{1}{2^n}}\right) = \Theta\left(\frac{n}{\varepsilon}\right)$

sample complexity is m^2

so $\Theta\left(\frac{n^2}{\varepsilon^2}\right)$

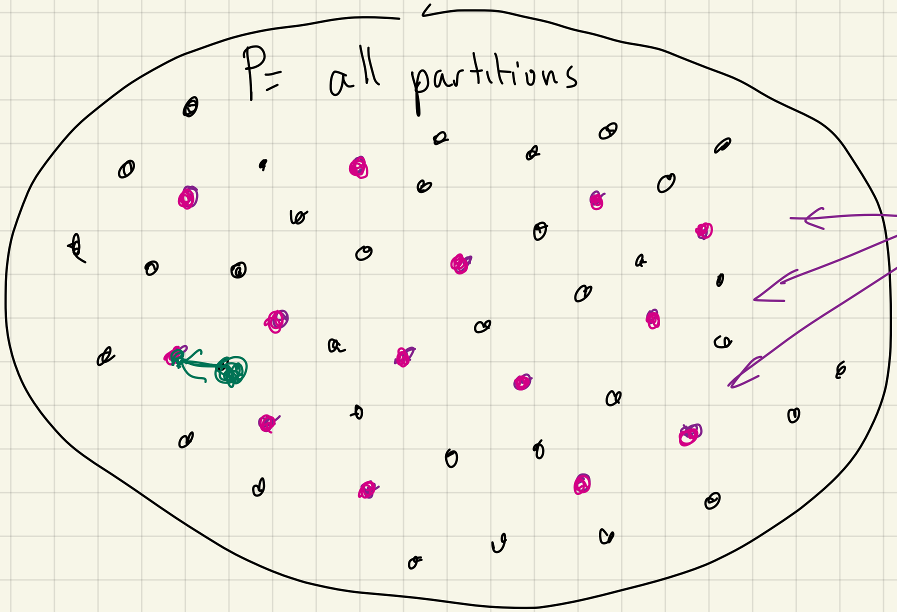
not sublinear

Do we really need a union bnd?

Do we need to try all partitions?

(or can we find few "representative" partitions that are close to all partitions?)

Plan: Consider small set of representatives



$|P| = 2^n$

R = purple points are "representatives"

- $R \subseteq P$
- every member of P is "close" to some member of R

Useful R satisfies:

- $\forall\ p \in P \quad \exists\ r \in R$ s.t. $\text{dist}(p, r) \leq \varepsilon$
- $|R|$ is small (hopefully $|R| << |P|$)

(1) if $p \in P$ that is a bipartition of G then $\exists r \in R$ s.t. r has few violations

(2) if $\forall p \in P$, p far from bipartition then since $R \subseteq P$, all r are far too

Plan:

Find "representative" partitions s.t. all partitions in P are $\frac{\varepsilon}{2}$-close to some representative.

- if G ε-far from bipartite then
  $\forall$ partitions $\geq \varepsilon n^2$ violating edges

  $\Rightarrow$ $\forall$ representative partitions have $\geq \varepsilon n^2$ violating edges, since $R \subseteq P$

- if G bipartite then
  $\exists$ partition with 0 violating edges

  $\Rightarrow$ so $\exists$ representative partition with $< 0 + \frac{\varepsilon}{2} n^2$ violating edges
  $= \frac{\varepsilon}{2} n^2$

Algorithm 1

1. pick U, U' randomly from V

$U = \Theta\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)$ nodes

$U' = \Theta\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\right)$ nodes

Used to define "R" the set of partitions

Used to test partitions!

think of $U' = \{u_1, v_1, u_2 v_2 \ldots\}$

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

If U not bipartite, FAIL $\leftarrow O(|U|^2)$ queries

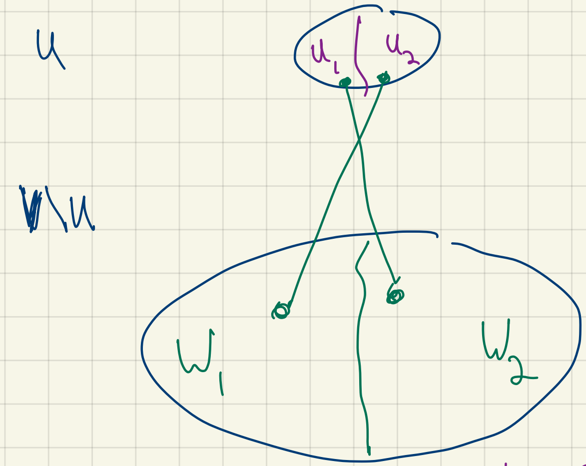
2. $\forall$ partitions of U into U_1, U_2: (only consider those that are bipartitions of U)

$2^{|U|}$ of these

- define oracle (see below) which partitions graph into Z_1, Z_2 based on U_1, U_2
- $\forall\ u \in U'$ call oracle to see if u in Z_1 or Z_2
- Count $\#\{(u,v) \in P$ that violate $Z_1, Z_2\}$

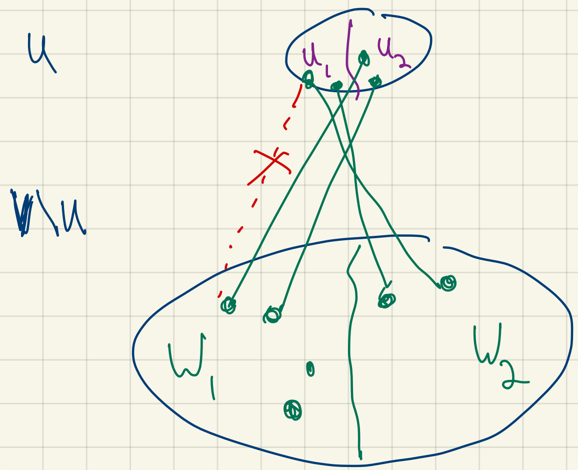
  if fraction $\leq 3/4\,\varepsilon$ output PASS + halt $\Leftarrow$ why pass if > 0 violations? since we don't check all partitions

  else continue to next partition



3. FAIL

Given partition of U into U_1, U_2, define ORACLE to partition whole graph:



Query: node v

Oracle answer: Z_1 or Z_2 or "bad partition"

Oracle algorithm:

output Z_1 if

$v \in U_1$

v has nbr in U_2 but not in U_1

v has no nbr in either U_1 or U_2 ??

$Z_1 = U_1 \cup W_1$

$Z_2 = U_2 \cup W_2$

else output Z_2 if

$v \in U_2$

v has nbr in U_1 but not U_2

else output "bad partition" ← only reach if have nbr to U_1 & U_2

Runtime: $O(|U|)$ per query

Algorithm 1

1. pick U, U' randomly from V

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

   $U' = \Theta(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon})$ nodes — used to define random edges: pair off $U' = \{u_1, v_1, u_2, v_2, \dots\}$ to $P = \{(u_1,v_1), (u_2,v_2), \dots\}$ pairs

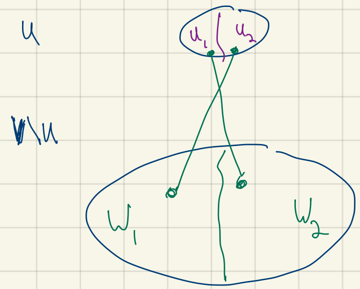
   If U not bipartite, FAIL

2. $\forall$ partitions of U into U_1, U_2:
   - define oracle (see below) which partitions graph into Z_1, Z_2 based on U_1, U_2
   - $\forall u \in U'$ call oracle to see if u in Z_1 or Z_2
   - count $\#\{(u,v) \in P$ that violate $Z_1, Z_2\}$

     if fraction $\le 3/4\,\varepsilon$ output PASS & halt

     else continue to next partition

3. FAIL



Query Complexity:

$$\frac{1}{\varepsilon}\log\frac{1}{\varepsilon} \cdot \frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon} = O\left(\frac{1}{\varepsilon^3}\log^2\frac{1}{\varepsilon}\right)$$

Time Complexity:

$$O\left(2^{\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}} \times \dots\right)$$

no dependence on n

can improve dependence on ε

Behavior: need to show that if G bipartite, likely to pass & if G ε-far from bipartite, likely to fail

if G is ε-far:

all partitions Z_1, Z_2, including those tested by algorithm, have $> \varepsilon n^2$ violating edges

$\forall Z_1, Z_2$ $\Pr[\text{fraction of violating edges in } P \text{ is} \le \frac{3}{4}\varepsilon n^2] \ll \frac{1}{8 \cdot 2^{|U|}}$ ((Chernoff bnd))

$\Pr[\text{PASS}] = \Pr[\text{any } Z_1, Z_2 \text{ passes}] \le 2^{|U|} \cdot \frac{1}{8 \cdot 2^{|U|}} \ll \frac{1}{8}$

Algorithm 1

1. pick U, U' randomly from V

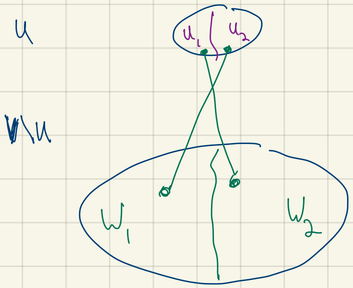
$U = \Theta\left(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon}\right)$ nodes — used to define set of partitions

$U' = \Theta\left(\frac{1}{\varepsilon^2}\log\frac{1}{\varepsilon}\right)$ nodes — used to define random edges: pair off $U' = \{u_1, v_1, u_2, v_2, \ldots\}$ to $P = \{(u_1,v_1), (u_2,v_2), \ldots\}$ pairs

If U not bipartite, FAIL

2. $\forall$ partitions of U into U_1, U_2:
   - define oracle (see below) which partitions graph into Z_1, Z_2 based on U_1, U_2
   - $\forall\, u \in U'$ call oracle to see if u in Z_1 or Z_2
   - count $\#\{(u,v) \in P$ that violate $Z_1, Z_2\}$
     if fraction $\leq 3/4\,\varepsilon$ output PASS + halt
     else continue to next partition

3. FAIL



if G is bipartite:

does it pass?

Let (Y_1, Y_2) be bipartite partition.

$\#$ violating edges $= 0$

For sample U, partition according to Y_1, Y_2:

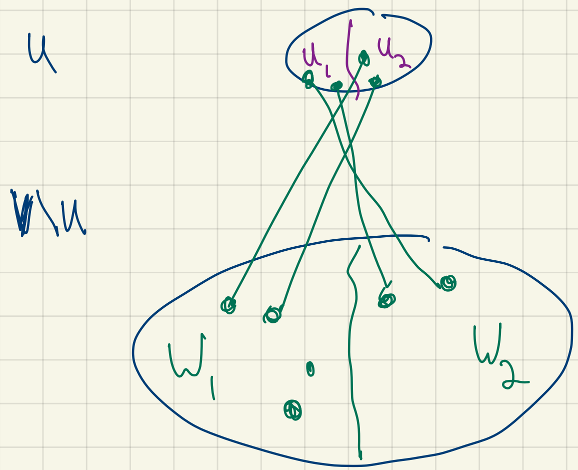
$$\left.\begin{aligned} U_1 &\leftarrow U \cap Y_1 \\ U_2 &\leftarrow U \cap Y_2 \end{aligned}\right\} \text{partition of } U \text{ according to } (Y_1, Y_2)$$

Use (U_1, U_2) to $\underbrace{\text{partition}}_{\text{thought process}}$ V into $W_1^{U_1 U_2}$ $W_2^{U_1 U_2}$

Question: how similar is (Y_1, Y_2) to $\underbrace{(W_1^{U_1,U_2}, W_2^{U_1,U_2})}_{\text{get from oracle on } U_1, U_2}$?

how many extra violating edges?

Given partition of U into U_1, U_2, define ORACLE to partition whole graph:



$Z_1 = U_1 \cup W_1$

$Z_2 = U_2 \cup W_2$

Query: node v

Oracle answer: Z_1 or Z_2 or "bad partition"

Oracle algorithm:

output Z_1 if

$v \in U_1$

v has nbr in U_2 but not U_1

v has no nbr in U_1 or U_2

only place where Y_1, Y_2 can differ

else output Z_2 if

$v \in U_2$

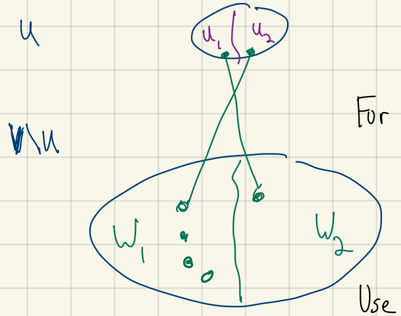
v has nbr in U_1 but not U_2

else output "bad partition"

Runtime: $O(|U|)$ per query

$\forall$ partitions of U into U_1, U_2:

• induce partition $(\overset{(Z_1,}{U_1 \cup W_1}, \overset{Z_2)}{U_2 \cup W_2})$ on whole graph:



Let (Y_1, Y_2) be bipartite partition.

violating edges $= 0$

For sample U, partition according to Y_1, Y_2:

$U_1 \leftarrow U \cap Y_1$

$U_2 \leftarrow U \cap Y_2$

Use U_1, U_2 to partition V: $W_1^{U_1 U_2}$, $W_2^{U_1 U_2}$

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

$\leq \underbrace{0}_{\substack{\text{\# violating edges} \\ \text{in } (Y_1, Y_2)}} + \underbrace{\text{\# edges adj to any } v \text{ that has no nbr in } U}_{\text{divide into 2 groups!}}$

$A = \{ v \text{ s.t. } \deg v < \frac{\varepsilon}{4} n \}$ "small degree"

$B = V \setminus A$ "high degree"

$\leq \frac{\varepsilon n}{4} \cdot n + n \cdot \boxed{}$

max degree of $v \in A$

upper bnd on $|A|$

max deg of $v \in B$

upper bnd on $|B|$

need to find a good bound

recall: U is random sample, $|U| = \Theta(\frac{1}{\varepsilon}\log\frac{1}{\varepsilon})$

$B_U = \{ v$ s.t. $\deg(v) \geq \frac{\varepsilon}{4} n$ & v has no nbr in $U \}$

<u>Lemma</u> $\Pr_U\left[|B_U| \leq \frac{\varepsilon}{4} n \right] \geq 7/8$

<u>Pf</u> $\forall$ v of $\deg \geq \frac{\varepsilon}{4} n$ set $\sigma_v \leftarrow \begin{cases} 1 & \text{if } v \in B \\ 0 & \text{o.w.} \end{cases}$

$E_U[\sigma_v] = \Pr_U[\sigma_v = 1] = \left(\Pr[i^{th} \text{ node of } U \text{ isn't nbr of } v] \right)^{|U|}$

$\leq \left(1 - \frac{\varepsilon}{4}\right)^{|U|} \leq \left(1 - \frac{\varepsilon}{4}\right)^{\frac{4}{\varepsilon} \cdot \log 32/\varepsilon} \leq \frac{\varepsilon}{32}$ $\leftarrow$ by picking right constants $|U| = \frac{4}{\varepsilon} \log 32/\varepsilon$

$E\left[\sum_{\substack{v \text{ s.t.} \\ \deg v \geq \frac{\varepsilon}{4} n}} \sigma_v \right] \leq \frac{\varepsilon n}{32}$ so $\Pr\left[|B_U| = \sum \sigma_v \geq \underbrace{\frac{8 \varepsilon n}{32}}_{\varepsilon n/4} \right] \leq \frac{1}{8}$ by Markov $\neq$ ☒

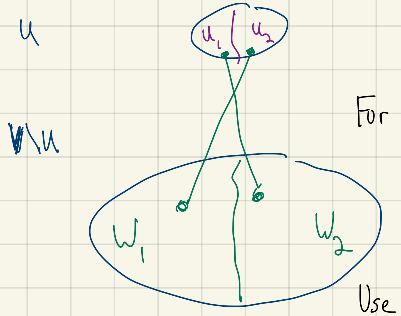
$\forall$ partitions of U into U_1, U_2:

- induce partition $\overset{(Z_1, Z_2)}{(U_1 \cup W_1, U_2 \cup W_2)}$ on whole graph:

  output Z_1 if
  - $v \in U_1$
  - v has nbr in U_2 but not U_1
  - v has no nbr in U_1 or U_2

  else output Z_2 if
  - $v \in U_2$
  - v has nbr in U_1 but not U_2

  else output "bad partition"



Let (Y_1, Y_2) be bipartite partition.

$\#$ violating edges $= 0$

For sample U, partition according to Y_1, Y_2:

$U_1 \leftarrow U \cap Y_1$

$U_2 \leftarrow U \cap Y_2$

Use U_1, U_2 to partition V: $W_1^{U_1 U_2}, W_2^{U_1 U_2}$

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

$\le \underbrace{0}_{\#\text{ violating edges in } (Y_1, Y_2)} + \overbrace{\#\text{ edges adjacent to any } v \text{ with no nbr in } U}^{\#\text{edges that can differ between } (Y_1, Y_2) \text{ \& } (W_1^{U_1 U_2}, W_2^{U_1 U_2})}$

divide into two cases:

$A = \{ v \text{ s.t. } \deg(v) < \frac{\varepsilon}{4} n \}$ "small degree"

$B = \{ v \text{ s.t. } \deg(v) \ge \frac{\varepsilon}{4} n \}$ "high degree"

$\le \frac{\varepsilon \cdot n}{4} \cdot n + n \cdot \frac{\varepsilon n}{4} \le \frac{\varepsilon n^2}{2}$

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

n: upper bnd on $|A|$

n: max deg of $v \in B$

$\frac{\varepsilon n}{4}$: upper bnd on $|B|$

$\Rightarrow E[\text{fraction of } (u,v) \in P \text{ violating } W_1^{U_1 U_2} W_2^{U_1 U_2}] \le \frac{\varepsilon}{2}$

So $\Pr[$ " " " " $\ge \frac{3\varepsilon}{4}] << 1/8$ by Chernoff.

recall: U is random sample, $|U| = \Theta(\frac{1}{\varepsilon} \log \frac{1}{\varepsilon})$

$B_U = \{ v$ s.t. $\deg(v) \geq \frac{\varepsilon}{4} n$ & v has no nbr in $U \}$

$\Rightarrow$ Pr[output fail]

$$= \underbrace{\Pr[\text{output fail} \mid |B_U| > \tfrac{\varepsilon}{4} n]}_{\leq 1} \cdot \underbrace{\Pr[|B_U| > (\varepsilon/4) n]}_{\leq 1/8}$$

$$+ \underbrace{\Pr[\text{output fail} \mid |B_U| \leq \tfrac{\varepsilon}{4} n]}_{\leq 1/8} \cdot \underbrace{\Pr[|B_U| \leq (\varepsilon/4) n]}_{\leq 1}$$

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

Comments

Can improve runtime to poly$(1/\varepsilon)$

Proposed testing algorithm actually works

In adjacency list model (sparse graphs) need

$\Omega(\sqrt{n})$ queries,

Why more ?

finer grain distinction

dense model: bipartite vs. $\varepsilon \cdot n^2$ edges need to be removed

sparse model: bipartite vs. $\varepsilon \cdot \Delta \cdot n$ edges need to be r

Other problems: Partition Properties

e.g. Max cut

can we partition into k $\approx$ equal sized groups $1 \ldots k$ s.t. fraction of edges between groups $i \neq j$ $\approx a_{ij}$ ← inputs to problem

similar oracle-based algorithms:

<u>Maxcut</u>: pick random sample S

$\forall$ partitions of S into (S_1, S_2), create oracle:

if $v \in V \setminus S$ has more edges to S_2 than S_1

put in W_1

else " " W_2

then estimate # edges between $(W_1 \cup S_1)$ & $(W_2 \cup S_2)$

Output max value