Lecture 7 (part II)

Testing dense graphs

- bipartiteness
Adjacency Matrix model

$G$ represented by matrix $A$

can query $A$ in one step

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

Distance from property $P$:

def $G$ is $\varepsilon$-far from $P$ if must change $\geq \varepsilon \cdot n^2$

terms in $A$ to turn $G$ into member of $P$

Testing "sparse" properties:

all graphs are $\varepsilon$-close to connected in this model

$\Rightarrow$ trivial tester outputs "PASS" w/o looking at graph
<table>
<thead>
<tr>
<th>Graph type</th>
<th>max degree</th>
<th>natural representation</th>
<th>notion of distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Previously</td>
<td>sparse</td>
<td>$\Delta$</td>
<td>adjacency list</td>
</tr>
<tr>
<td>Now</td>
<td>dense</td>
<td>$\Omega$</td>
<td>adjacency matrix</td>
</tr>
</tbody>
</table>

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.

\[\forall e \in E \quad \text{s.t.} \quad u, v \in V_1 \text{ or } u, v \in V_2\]

not bipartite \(\implies\) \(\forall (V_1, V_2) \in \text{"violating edge"}\)

\(\varepsilon\)-far from bipartite: (definition)

- must remove \(\geq \varepsilon \cdot n^2\) edges to make bipartite.
- \(\forall\) partitions \(V = (V_1, V_2)\), \(\geq \varepsilon \cdot n^2\) violating edges
Testing Algorithms:

- Testing exact bipartiteness:
  
  e.g. BFS

- Proposed testing algorithm:

  - Pick sample of nodes of size \( O(\frac{\varepsilon}{\Delta} \log \frac{1}{\Delta}) \)
  - Consider induced graph on sample
  - If bipartite, output PASS
    
    else output FAIL

  e.g. BFS

Goldreich, Goldwasser, Ron

\[ G \]

Ignore nodes not in sample
Ignore edge st.
\[ \geq \varepsilon \] end pt. is not in sample

This actually works!!
A first attempt at a proof?

if \( G \) bipartite, induced graph is bipartite, so algorithm Passes

if \( G \) \( \epsilon \)-far from bipartite:

must remove \( \epsilon n^2 \) edges to make it bipartite

equivalently:

A partition \( V_1, V_2 \) have \( \geq \epsilon n^2 \) violating edges (\( \geq \epsilon \) fraction of slots in adjacency matrix)

\[ \Rightarrow \ \forall (V_1, V_2) \] a sample of edges of size \( \geq \theta(\frac{1}{\epsilon \log \frac{1}{\epsilon}}) \)

hits a \((V_1, V_2)\) violating edge with prob \( \geq 1 - e^{-c \cdot \log \frac{1}{\epsilon}} = 1 - e^{-\log \frac{1}{\epsilon} = 1 - \frac{1}{\epsilon}} \) (set \( c = 1 \))

Great! ?

need to hit violating edge for every partition

how is this an algorithm?

no edge violates all partitions