Lecture 10 Nov. 22-74

Learning a hidden graph (Complex model)

(•) If we restrict all the complexes in complex model to be of size "2", then we can model the group testing as a learning graph problem.

Problem There is a graph G defined on $[1, n]$. The aim is to reconstruct the graph by using the so-called edge-detecting queries i.e. to ask: if $S \subseteq [1, n]$ (induces a clique) contains an edge of G. This is equivalent to ask if S contains any 2-subset (edge) in $E(G)$.

For example, if $n = 8$ and G contains only one edge $\{3, 7\}$, then we would like to find it by using adaptive algorithm as fast as possible.

(•) Note that in learning graph model, we don't know the number of edges in G in general. This type of group testing is known as a competitive model or competitive group testing.

A solution to the above problem.

Q_1: Is $[1,8]$ induces an edge (of G)? A_1: Yes.

Q_2: Is $[1,4]$ induces an edge (of G)? A_2: No.

Q_3: Is $[5,8]$ induces an edge (of G)? A_3: No.

Q_4: Is $[1,6]$ induces an edge (of G)? A_4: No.

Q_5: Is $[1,7]$ induces an edge (of G)? A_5: Yes.

Q_6: Is $\{1,2,7\}$ induces an edge (of G)? A_6: No.

Q_7: Is $\{3,7\}$ induces an edge (of G)? A_7: Yes.

Q_8: Is $[1,8]$ induces an edge (after deleting $\{3,7\}$)? A_8: No.

STOP!

So, it takes 8 queries to reconstruct G. Is this a good algorithm by using splitting idea? The following theorem gives a good lower bound.

<u>Theorem 1.</u> If G is a graph of order n and size m, then $\left\lceil \log\left(\sum_{i=0}^{m} \binom{\binom{n}{2}}{i} \right) \right\rceil$ queries are required to reconstruct G.

(Using adaptive algorithm)

Proof. It is not easy at all. But, the idea comes from that we can find one edge at a time and the answer is complete after all m edges are found.

($\circ$) $n=8$, $m=1$, Lower bound is $\left\lceil \log_2\left(\binom{28}{0}+\binom{28}{1}\right)\right\rceil = 5$.

($\circ\circ$) The work by Angluin and Chen in J. Machine Learning Research, 7, 2006, shows that "we can find an edge by using $2\lceil \log_2 n \rceil$ queries". (In this case, $m \geq 1$ is known.)

Exercise 1 Can you find an algorithm with this number of tests? (to detect one edge) Or, can you do a better job?

($\circ$) $2\lceil \log_2 n \rceil$ is very close to a good upper bound.

$$\log_2\left(\binom{\binom{n}{2}}{1}\right) = \log_2 \frac{n(n-1)}{2} = \log_2 n(n-1) - \log_2 2$$

$$= \log_2 n + \log_2 (n-1) - 1 \approx 2\log_2 n - 1 \approx 2\lceil \log_2 n \rceil .$$

($\circ\circ\circ$) Angluin and Chen also consider $\|G\| = m$ and they can detect all the m edges by using at most about $12 m \log_2 n$ queries (J. Comput. Syst. Sci. 74, 2008).

A better idea for $m > 1$.

For example, $D = \{15, 38, 18\}$, ⓪. $\{1,2,3,4,5,6,7,8\}$, Yes.

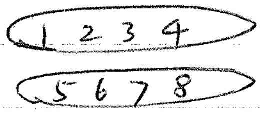
① $\{1,2,3,4\}$, No. ①′ $\{5,6,7,8\}$, No.

②. $\{1,2,3,4,5,6\}$, Yes

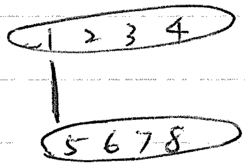
③ $\{1,2,3,4,5\}$, Yes. ⇒ Find 15 (③′)

④ $\{1,2,3,4,7,8\}$, Yes.

⑤ $\{1,2,3,4,7\}$, No. ⇒ Find 81 and 83.



After ①, ①′ (Independent set)

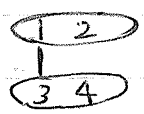


After ③′

(◦) No edges in a set S, put S as an independent set.

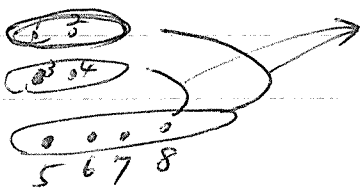
(◦◦) If $D = \{13, 38, 18\}$, then the answer of ① will be Yes.

$\{1,2,3,4\}$, Yes. ⇒ $\underline{\{1,2\}, \text{No.}}$ ⇒ $\{1,2,3\}$, Yes. ⇒ $\{1,3\}$, Yes.



$\{3,4\}$, No.

$\{1,2,4\}$ No. ⇒ ①′ ⇒



Edges are here!

A general result by Angluin and Chen.

<u>Theorem</u> Let $0<\varepsilon<2$ and $m=n^{2-\varepsilon}$. Then, $\varepsilon m(\log_2 n - 2)$ queries are required to identify (reconstruct) a graph G of order n and size m.

(o) ε 越大, m 越小。

<u>Theorem</u> (張惠蘭, 施智懷, Fu)

Let G be a graph of order n with m edges. Then, G can be reconstructed by using $m(\log_2 n + 4) + \alpha'(G)(\log_2 n + 5) + 1$ queries. ($\alpha'(G)$ is the maximum size of matching in G.)

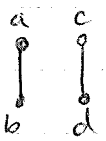
<u>Algorithm</u>

Step 1 Find a maximal matching M and an independent set I.

(Find an edge and remove its incident vertices!)

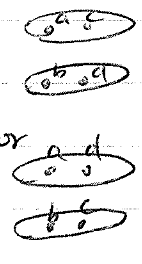
Step 2 Decompose $\langle [1,n] \setminus I \rangle$ into bipartite subgraphs.

$\Rightarrow$ add one edge



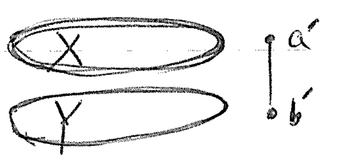
$\{a,c\},\{b,d\}$ No. or $\{a,d\},\{b,c\}$ No

$\Rightarrow$



or

$\Rightarrow$



$X\cup\{a'\} = 0 = Y\cup\{b'\}$

or $X\cup\{b'\} = 0 = Y\cup\{a'\}$.

(0) If $X \cup \{a'\} = 1$ or $Y \cup \{b'\} = 1$, or the other way around,

we find the edge and delete the edge.

Step 3 Identify the edges in bipartite graph.

Step 4 Identify the edge between $v \in [1, n] \setminus I$ and I.

After some extra effort, the upper bound was improved to

$\underline{\underline{m \log_2 n + 10m + 3n}}$ which is much better than $12 m \log_2 n$.