# Lecture Note 10

April 30, 2018

# 1 Social Networks

- Detecting Triangles
- Finding dense subgraphs
- Community detection

# 2 Find Dense Component in Graph

## 2.1 Problem Definition

Given a graph $G(V, E)$, find a subset $S \subseteq V$ such that the ratio $\frac{|E(S)|}{|S|}$ is maximized, where $E(S) = \{(u, v)|(u, v) \in E, u, v \in S\}$. Or in other words:

$$S = \arg\max_{S \subseteq V} \frac{|E(S)|}{|S|}.$$

## 2.2 Charikar Greedy Algorithm

This problem can be solved optimally in polynomial time, but it is too slow on large graphs. If we do not need exact solution, there is a 2-approx greedy algorithm that runs in linear time. Ref: Moses Charikar, APPROX 2000, link

### 2.2.1 Algorithm

The algorithm maintains a subset $S$ of vertices. Initially $S \leftarrow V$. In each iteration, the algorithm identifies $i_{\min}$, the vertex of minimum degree in the subgraph induced by $S$. The algorithm removes $i_{\min}$ from the set $S$ and moves on to the next iteration. The algorithm stops when the set $S$ is empty. Of all the sets $S$ constructed during the execution of the algorithm, the set $S$ maximizing density is returned as the output of the algorithm.

### 2.2.2 Pseudo code

- $S \leftarrow V$
- $\mathsf{SOL} = V$
- While $S \neq \emptyset$
  - \$v← \$ a vertices in $S$ whose degree in $G[S]$ (subgraph induced by $S$) is minimized

– Remove $v$ from $S$
– If density of $G[S] > G[\mathsf{SOL}]$
  * $\mathsf{SOL} \leftarrow S$

#### 2.2.3 Analysis

- Let $c^*$ be the first node we delete from $S^*$, degree of this node $\geq \lambda$
- $\Rightarrow s$ nodes have deg $\geq \lambda$
- $\Rightarrow$ # edges $\geq \frac{\lambda s}{2}$
- $\Rightarrow$ density $\geq \frac{\lambda}{2}$

### 2.3 LP formulation

$x_u$ is an indicator variable, 0 indicating it not in $S$, $\frac{1}{|S|}$ if it is in $S$. $y_e$ indicating an edge $e = (u, v)$ in $S$ or not. It is $\frac{1}{|S|}$ if $u, v \in S$, and 0 otherwise. The target function then is

$$\frac{1}{|S|} \cdot |\{\text{ of edges in } S\}| = \frac{|E(S)|}{|S|}$$

#### 2.3.1 LP formulation

$$\begin{aligned} &\max \sum_e y_e \\ \text{Subject to} \quad & y_e \leq x_u, \;\; y_e \leq x_v \quad \forall e = (u, v) \\ &\sum_{v \in V} x_v = 1 \\ &0 \leq y_e \leq 1 \\ &0 \leq x_v \leq 1 \end{aligned}$$

## 3 Counting Triangles

Count the number of triangles in a graph

### 3.1 Number of Triangles in a Random graph

- $m$ edges, $n$ nodes.
- Probability that there is an edge between a certain pair of vertices:
$$p = \frac{m}{\frac{1}{2} \cdot n(n-1)} = \left(\frac{2m}{n^2}\right)$$
- Expected number of triangles:
$$\binom{n}{3} \cdot p^3 \simeq (\frac{1}{6} n^3) \cdot p^3$$
- Plugging in $p$:
$$\frac{1}{6} \cdot n^3 \cdot \frac{8m^3}{n^6} = \frac{4}{3} \left(\frac{m}{n}\right)^3$$

## 3.2 Algorithm

- $\mathsf{count} \leftarrow 0$
- For every edge $(u, v)$, suppose $\mathsf{deg}(u) < \mathsf{deg}(v)$ with out loss of generality
  - For $w$ in neighbor list of $u$
    - If $w$ is a neighbor of $v$, then $\mathsf{count} + = 1$

## 3.3 Arboricity

Arboricity is a measure of how sparse a graph is. The arboricity $\alpha$ of a graph is defined as

$$\max_{S \subseteq V} \lceil \frac{E(S)}{|S| - 1} \rceil.$$

Another closely related concept is degeneracy, which is defined as the smallest value $d$ such that every subgraph has a vertex of degree at most $d$. For every graph, $\alpha \leq d \leq 2\alpha + 1$, so degeneracy and arboricity are of the same order, or $d = O(\alpha)$.

For social graph of even millions of nodes, arboricity is pretty small, about 100-300. As a special case, any planar graph has arboricity at most 3. One fact about arboricity is as follows:

$$\sum_{(u,v) \in E} \min(\mathsf{deg}(u), \mathsf{deg}(v)) = O(\alpha \cdot m).$$

## 3.4 Time Analysis

For every edge $(u, v)$, the time is $\min(\mathsf{deg}(u), \mathsf{deg}(v))$. So the total time complexity of the algorithm above is

$$\sum_{(u,v) \in E} \min(\mathsf{deg}(u), \mathsf{deg}(v)) = O(\alpha \cdot m).$$