Local Algorithms for Finding Interesting Individuals in Large Networks

Mickey Brautbar, Michael Kearns

Computer and Information Science
University of Pennsylvania
Innovations in Computer Science 2010

Structural Properties of Social Networks

- Nodes with high degree.
- Nodes with high clustering coefficient.
- Nodes with high betweenness centrality (measures how many shortest paths between other nodes are passing through the given node).
- Small diameter.
- Many other structural properties...

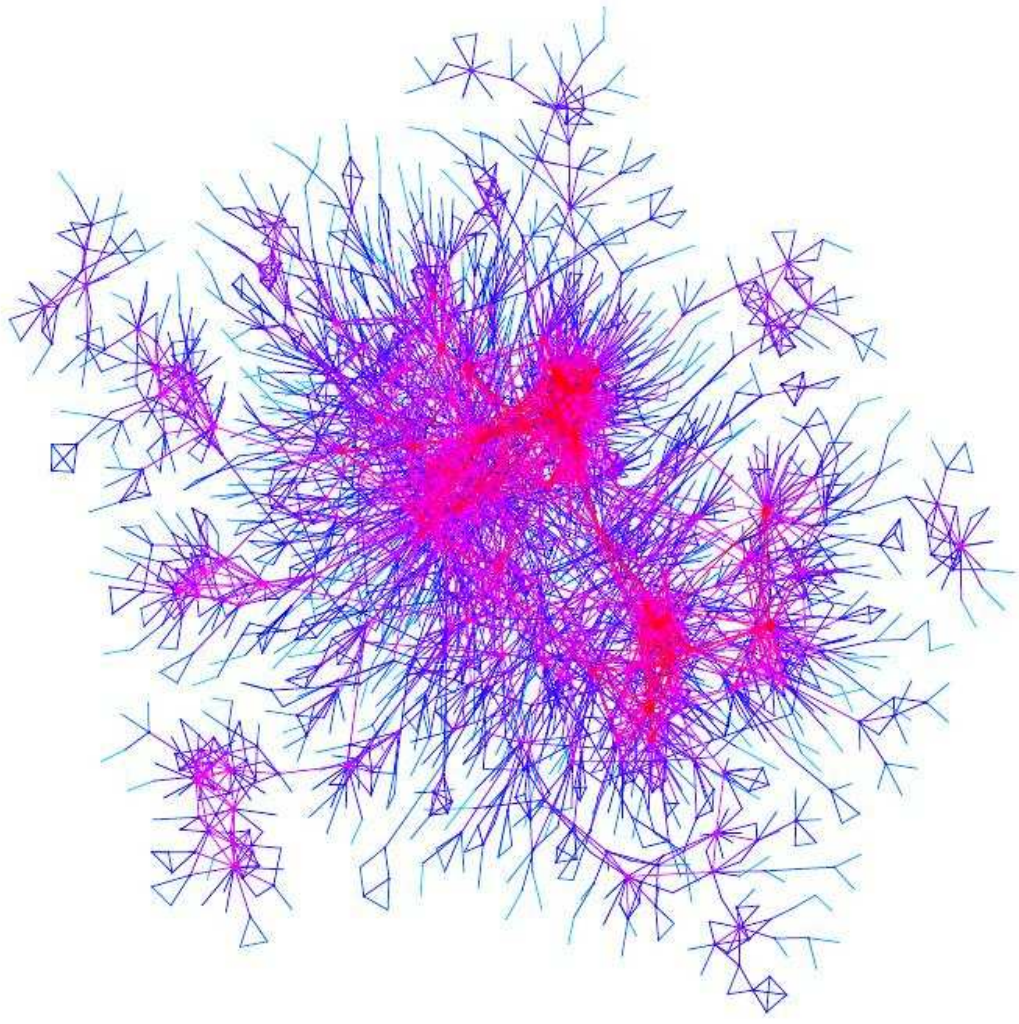

FIGURE 20. A subgraph of the Hollywood graph.

Figure: Taken from 'Complex Graphs and Networks' by Chung and Lu.

The Algorithmic Side

- Question 1: If the network contains the structure at hand, how hard is it to find that structure?
- Question 2: Is it substantially easier to find such a structure in a social network than finding such a structure in an arbitrary network (that contains the structure)?
- Potential applications.

Example: Marketing of new product of small budget companies
Find a node with high degree in an online social network and pay that node to add a link to your new product from his online profile.

What Kind of Query Model Should We Use?

Facebook as a model

- As a user of Facebook I am lacking a central view of the Facebook network.
- The edges in the network are the connections between friends.
- How hard would it be for me to find a user with many friends?
- I can make a **Crawl** query - go to the profile of my friend and check his degree.
- I can make a **Jump** query by using the *Friend's Finder* option of Facebook. Typing a random name in *Friend's Finder* will return a user with that profile (if one exists).

The Model

- At a given user we know his degree and also the identity of his immediate neighbors.
- We can make a **Crawl** query to move to a specified neighbor of the node at hand.
- Or we can make a **Jump** query to move to a uniformly at random chosen node from the network.

Goal

- In this talk we shall focus on the high degree structure and also the high clustering coefficient structure.

Definition (approximation of high degree)

Given a parameter $0 \leq \alpha \leq 1$ and a network with n nodes, we say that v gives an approximation ratio of $n^{1-\alpha}$ if $d^* \leq \text{degree}(v) \cdot n^{1-\alpha}$.

- For $\alpha = 0$, any node works (in a connected network).
- For $\alpha = 1$, only a node with the highest in the network is a valid answer

- Goal: Find such a vertex v in **sublinear time** in the size of the network.
- Answer will depend on type of network considered. Three network categories: arbitrary networks, power law networks, and preferential attachment networks.

High Degree Structure - Arbitrary Connected Network

- Question 1: If the network contains the structure at hand, how hard is it to find that structure?
- Answer- lower bound: Let G be a $k-$edge connected network, for a fixed k. Let $0 < \beta < \frac{1}{2}$. Then $\Omega(n^{\beta})$ Jump and Crawl queries are necessary in order to find a node that gives an approximation ratio of $O(n^{1-\beta})$.
- Proof idea: In the line-clique graph n^{β} Jump queries will leave us $n^{1-\beta}$ distance away from the clique subnetwork, so even with Crawl queries the algorithm will see only degree 2 nodes. However, the highest degree is $n^{1-\beta}$.

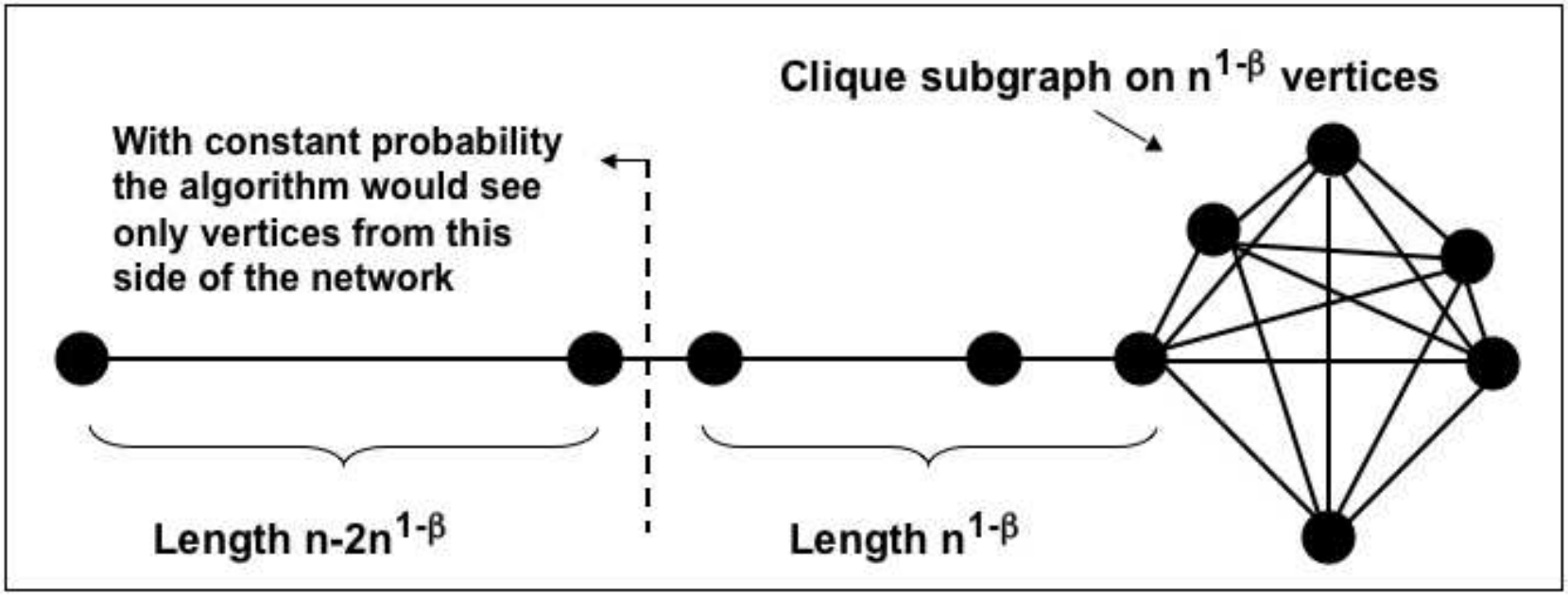

Figure: A line-clique network

High Degree Structure - Arbitrary Connected Network

- Question 1: If the network contains the structure at hand, how hard is it to find that structure?
- Answer- upper bound: Let G be an arbitrary network and let $0 < \beta < 1$. Then there exists an algorithm that with $\tilde{O}(n^{\beta})$ Jump and Crawl queries finds a node which gives an approximation ratio of $n^{1-\beta}$, whp.
- Algorithm sketch:
  - Let v^* be a maximum degree node (with degree d^*).
  - If $d^* < n^{1-\beta}$ any node will do.
  - Else, with $\tilde{O}(\frac{n}{d^*})$ Jump queries a neighbor of v^* is found.
  - Unless such a node has already degree higher than $\frac{d^*}{n^{1-\beta}}$, by Crawling that node's immediate neighbors we find v^*.
  - Total number of queries is $\tilde{O}(n^{\beta})$.

High Degree Structure - Power Law Network with n Nodes

- Degree distribution follows a power law, namely, $prob(d) \propto d^{-\gamma}$, for $d = 1, 2, ..., t$.
- Question 2: Is it substantially easier to find such a structure in a social network than finding such a structure in an arbitrary network that contains the structure?
- Answer: Yes. Let G be a power Law network with an exponent $\gamma > 2$. Then there exist an algorithm that with $\tilde{O}(n^{\beta})$ Jump and Crawl queries finds a node which gives an approximation ratio of $O(n^{\frac{1}{\gamma} - \frac{\beta}{\gamma - 1}})$, whp.

Power Law Network with n Nodes - Cont.

Algorithm sketch: Take Jump $\tilde{O}(n^{\beta})$ Jump queries and return the node with the highest degree between those found.

Proof sketch:

- The probability of randomly sample a node of degree at least $n^{(\frac{\beta}{\gamma-1})}$ is $\theta(n^{-\beta})$ in a power law network with exponent γ.
- Highest degree in a power law network is $\theta(n^{\frac{1}{\gamma}})$.

High Degree Structure - Preferential Attachment Network

The PA process creates a network with n nodes in the following way:

- Start with a connected subgraph on m nodes.
- In each time step add a new node u and m new links between u and previously added nodes.
- $prob((u, v)) \propto degree(v)$.

Preferential Attachment Network - Some Useful Properties

- **Expected** degree distribution is a power law of exponent 3.
- The realized degree is close to its expected value for degrees smaller than $n^{\frac{1}{11}}$ (implied from a result by Chung and Lu).
- Maximum degree in a realization of the PA process is about $\sqrt{n}$ whp (Flaxman, Frieze, Fenner).
- Corollary - the previous result for power law networks holds for small degrees in the preferential attachment network : with $\tilde{O}(n^{\beta})$ queries achieves an approximation ratio of $O(n^{\frac{1}{2}-\frac{\beta}{2}})$.

High Degree Structure - Preferential Attachment Network

- We can do better using lazy random walks.
  - The Lazy Random Walk (LRW) stays put with probability $\frac{1}{2}$ and with probability $\frac{1}{2}$ uniformly crawl to a neighbor.
  - LRW is mixing fast to the stationary distribution on a PA network.
- Let G be generated using the preferential attachment process. Then, there exists an algorithm that with $\tilde{O}(n^{\beta})$ Crawl queries finds a node which gives an approximation ratio of $O(n^{\frac{1}{2}-\beta})$, whp.

Algorithm idea: Sample $\tilde{O}(n^{\beta})$ times from the degree distribution by running the LRW.

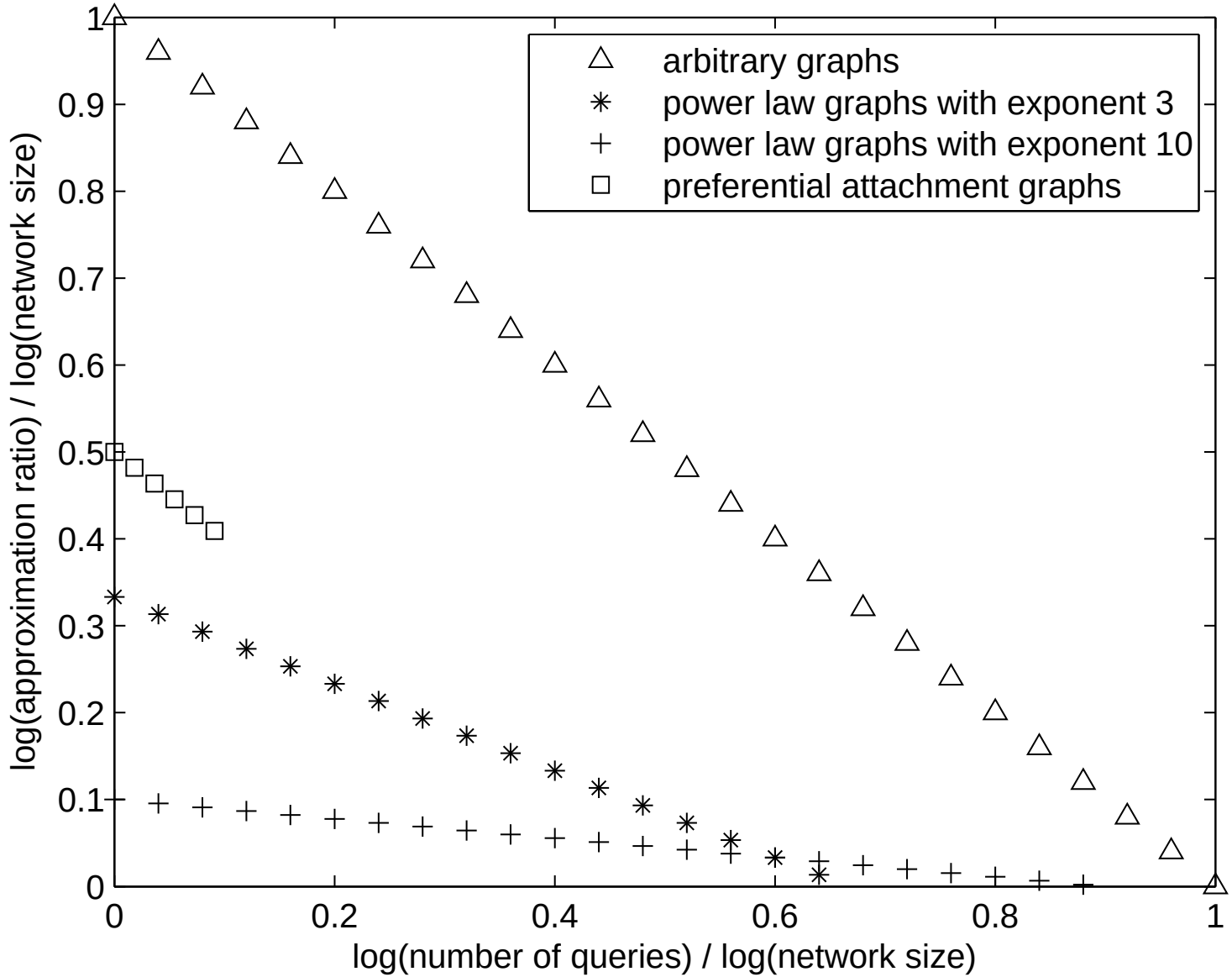

Figure: Achievable rates for all categories.

The Clustering Coefficient Structure

Definition (Clustering Coefficient)

Given a vertex v with degree d, the clustering coefficient (CC) of v is defined as $CC(v) = \frac{\text{number of triangles containing } v}{\binom{d}{2}}$.

Definition (approximation of Clustering Coefficient)

Given a graph and a degree value d, let v^* be the vertex with the highest CC among vertices of degree d or more. We say that v is a (α, d, ϵ)-approximation to the maximum CC if $degree(v) \geq \alpha \cdot d$ and $CC(v^*) \leq CC(v) + \epsilon$, for $0 < \alpha \leq 1$ and $0 < \epsilon < 1$.

$(1, n^{\beta}, 1/2)$ Approximation is Impossible with $n^{1-\beta}$ Queries

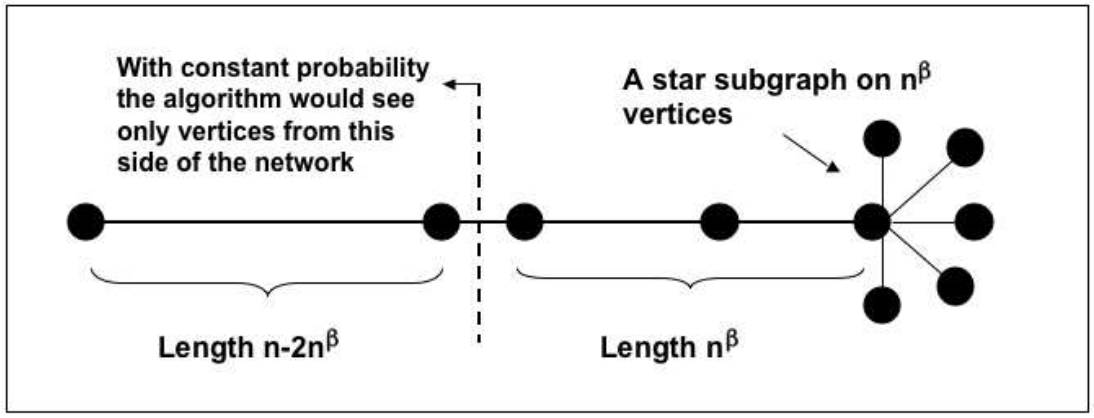

Figure: The line-star network

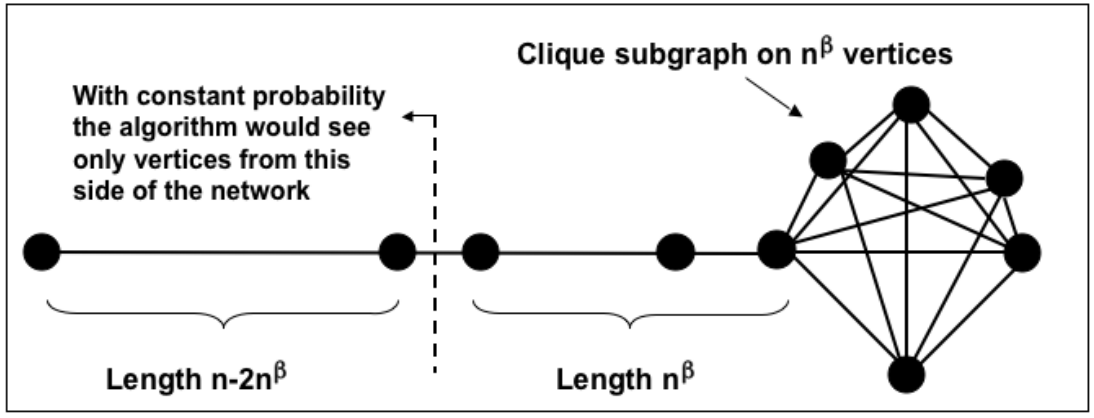

Figure: The line-clique network

The Clustering Coefficient Structure

Lower bound is tight if we allow logarithmic slack.

- There exists an algorithm that with $\tilde{O}(n^{1-\beta})$ Jump and Crawl queries returns a $(\frac{1}{\log n}, n^{\beta}, \epsilon)$ approximation to the maximum clustering coefficient, $\epsilon \longrightarrow 0$.
- Proof is quite involved - is given in the paper.
- For power law networks where the LRW is mixing fast, e.g. the PA network, we can do substantially better.
  There exists an algorithm that with $\tilde{O}(n^{1-2\beta})$ Jump and Crawl queries returns a $(\frac{1}{\log n}, n^{\beta}, \epsilon)$ approximation to the maximum clustering coefficient.
- If $\beta \geq \frac{1}{2}$ algorithm uses only a poly-logarithmic number of queries for achieving such an approximation.

Future Work

Using sublinear number of *Jump* and *Crawl* queries:

- Find two nodes that are diameter distant from each other.
- Find a node with high betweenness centrality measure (measures how many shortest paths between other nodes are passing through the given node).
- Other structural properties.
- Thank You