Game Theoretic Models for Social Network Analysis

Y. Narahari and Ramasuri Narayananam

Email ID: {hari,nrsuri}@csa.iisc.ernet.in
Electronic Commerce Laboratory
Department of Computer Science and Automation
Indian Institute of Science
Bangalore, India

28-March-2011
Outline of the Presentation

1. **Social Network Analysis: Quick Primer**
2. Foundational Concepts in Game Theory
3. Discovering Influential Individuals for Viral Marketing
4. Social Network Formation
5. Community Detection in Social Networks
6. Query Incentive Networks
7. Summary and To Probe Further
Social Networks: Introduction

Social networks are ubiquitous and have many applications:

- For targeted advertising
- Monetizing user activities on on-line communities
- Job finding through personal contacts
- Predicting future events
- E-commerce and e-business
- . . .

__________________________

Example 1: Web Graph

Nodes: Static web pages
Edges: Hyper-links

Example 2: Friendship Networks

Friendship Network

Nodes: Friends
Edges: Friendship
Reference: Moody 2001

Subgraph of Email Network

Nodes: Individuals
Edges: Email Communication
Reference: Schall 2009
Example 3: Weblog Networks

Nodes: Blogs
Edges: Links

Reference: Hurst 2007
Example 4: Co-authorship Networks

Nodes: Scientists and Edges: Co-authorship

Example 5: Citation Networks

Nodes: Journals and Edges: Citation

Reference: http://eigenfactor.org/
Social Networks - Definition

- **Social Network**: A social system made up of individuals and interactions among these individuals

- Represented using graphs
  - Nodes - Friends, Publications, Authors, Organizations, Blogs, etc.
  - Edges - Friendship, Citation, Co-authorship, Collaboration, Links, etc.

---

Social Network Analysis (SNA)

- Study of structural and communication patterns
  - degree distribution, density of edges, diameter of the network

- Two principal categories:
  - **Node/Edge Centric Analysis:**
    - Centrality measures such as degree, betweenness, stress, closeness
    - Anomaly detection
    - Link prediction, etc.
  - **Network Centric Analysis:**
    - Community detection
    - Graph visualization and summarization
    - Frequent subgraph discovery
    - Generative models, etc.

_____________________

Why is SNA Important?

- To understand complex connectivity and communication patterns among individuals in the network
- To determine the structure of networks
- To determine influential individuals in social networks
- To understand how social networks evolve
- To determine outliers in social networks
- To design effective viral marketing campaigns for targeted advertising
  . . .
What are Key Issues in SNA?

**Measures to Rank Nodes:**

- **Degree Centrality:** It is defined as the number of links incident upon a node.
- **Clustering Coefficient:** It measures how dense is the neighborhood of a node.
- **Between Centrality:** It is a measure of a node and a node that occurs on many shortest paths between other pairs of nodes has higher betweenness centrality.
- **Closeness Centrality:** It is defined as the mean shortest distance between a vertex and all other vertices reachable from it.
- **Eigenvector Centrality:** It assigns relative scores to all nodes in the network based on the principle that connections to high-scoring nodes contribute more to the score of the node in question than equal connections to low-scoring nodes.

...
What are Key Issues in SNA? (Cont.)

- **Diversity among Nodes:**
  - Nodes in the network might be having various connectivity patterns
  - Some nodes might be connected to high degree nodes, some others might be connected to bridge nodes, etc.
  - Determining diversity among the connectivity patterns of nodes is an interesting problem
What are Key Issues in SNA? (Cont.)

- **Link Prediction Problem:**
  - Given a snapshot of a social network, can we infer which new interactions among its members are likely to occur in the near future?

What are Key Issues in SNA? (Cont.)

- **Inferring Social Networks from Social Events:**
  - In the traditional link prediction problem, a snapshot of a social network is used as a starting point to predict (by means of graph-theoretic measures) the links that are likely to appear in the future.

  - Predicting the structure of a social network when the network itself is totally missing while some other information (such as interest group membership) regarding the nodes is available.

What are Key Issues in SNA? (Cont.)

**Viral Marketing:**

- With increasing popularity of online social networks, viral Marketing - the idea of exploiting social connectivity patterns of users to propagate awareness of products - has got significant attention.

- In viral marketing, within certain budget, typically we give free samples of products (or sufficient discounts on products) to certain set of influential individuals and these individuals in turn possibly recommend the product to their friends and so on.

- It is very challenging to determine a set of influential individuals, within certain budget, to maximize the volume of information cascade over the network.

Community Detection:

*Based on Link Structure in the Social Network:*
- Determining dense subgraphs in social graphs
- Graph partitioning
- Determining the best subgraph with maximum number of neighbors
- Overlapping community detection

*Based on Activities over the Social Network*
- Determine action communities in social networks
- Overlapping community detection


Query Incentive Networks:

- With growing number of online social communities, users pose queries to the network itself, rather than posing queries to a centralized system.
- At present, the concept of incentive based queries is used in various question-answer networks such as Yahoo! Answers, Orkuts Ask Friends, etc.
- In the above contexts, only the person who answers the query is rewarded, with no reward for the intermediaries. Since individuals are often rational and intelligent, they may not participate in answering the queries unless some kind of incentives are provided.
- It is also important to consider the quality of the answer to the query, when incentives are involved.
Determining the Implicit Social Hierarchy:

- Social stratification refers to the hierarchical classification of individuals based on power, position, and importance.

- The popularity of online social networks presents an opportunity to study social hierarchy for different types of large scale networks.

How to Address Issues in SNA?

Traditional Approaches
- Graph theoretic techniques
- Spectral methods
- Optimization techniques
- ...
How to Address Issues in SNA?

- **Traditional Approaches**
  - Graph theoretic techniques
  - Spectral methods
  - Optimization techniques
  - ...

- **Recent Advances**
  - Data mining and machine learning techniques
  - *Game theoretic techniques*
Next Part of the Talk

1. Social Network Analysis: Quick Primer
2. Foundational Concepts in Game Theory
3. Discovering Influential Individuals for Viral Marketing
4. Social Network Formation
5. Community Detection in Social Networks
6. Query Incentive Networks
7. Summary and To Probe Further
Contents to be Covered

- We first present a simple game theoretic model that brings out several aspects of viral marketing.
  

- We then bring out the challenges involved in viral marketing

- We discuss two standard models for propagation of influence in social networks and introduce the influence maximization problem

- Finally discuss a cooperative game theoretic model for influence maximization problem
Diffusion Game

- A game $\Gamma = (G, N)$ is induced by an undirected graph $G = (V, E)$, representing the underlying social network, and the set of agents $N$.

- The strategy space of each agent is the set of vertices $V$ in the graph, that is, each agent $i$ selects a single node and that node is colored in color $i$ at time 1. We call them *initial trend setters*.

- Note that if two or more agents select the same vertex at time 1 then that vertex becomes gray.

- *Diffusion Process*: At time $t + 1$, each white vertex that has neighbors colored in color $i$, but does not have neighbors colored in color $j$ for any $j \in N$, is colored in color $i$. 
A white vertex that has two neighbors colored by two distinct colors \(i, j \in N\) is colored gray. That is, we assume that if two agents compete for a user at the same time, they cancel out and the user is removed from the game.

The process continues until it reaches a fixed point, that is, all the remaining white vertices are unreachable due to gray vertices.

A strategy profile is a vector \(x = (x_1, x_2, \ldots, x_n)\), where \(x_i \in V\) is the initial vertex selected by agent \(i\). We also denote \(x_{-i} = (x_1, \ldots, x_{i-1}, x_{i+1}, \ldots, x_n)\).
Given a strategy profile $x \in V^n$, the utility of agent $i \in N$, denoted $U_i(x)$, is the number of nodes that are colored in color $i$ when the diffusion process terminates.

**Nash Equilibrium:** A strategy profile $x$ is a (pure strategy) Nash equilibrium of the game $\Gamma$ if an agent cannot benefit from unilaterally deviating to a diffusion strategy. That is, for every $i \in N$, and $x'_i \in V$, it holds that

$$U_i(x'_i, x_{-i}) \leq U_i(x)$$
Diffusion Game - Example

(a) Time 1.

(b) Time 2.

(c) Time 3, the process terminates.
If we can find a Nash equilibrium then we can often predict the behavior of the agents and the outcome of this competitive diffusion process.

**Theorem:** Every game $\Gamma = (G, N)$, where $D(G) \leq 2$ admits a Nash equilibrium. Furthermore, an equilibrium can be found in polynomial time.

**Theorem:** Let $N = \{1, 2\}$. There exists a graph $G$ with $D(G) = 3$ such that the game $\Gamma = (G, N)$ does not admit a Nash equilibrium.
Challenges in Viral Marketing

- Propagation of influence is a stochastic process, but not a deterministic process
- The number of individuals in the social network that are getting influenced by the initial trend setters is an expected quantity
- Viral marketing for single or multiple products
- There can possibly exist certain types of dependencies among these products
Motivating Example 1: Viral Marketing

- Social networks play a key role for the spread of an innovation or technology.
- We would like to market a new product that we hope will be adopted by a large fraction of the network.
- Which set of the individuals should we target for?
- Idea is to initially target a few influential individuals in the network who will recommend the product to other friends, and so on.
- A natural question is to find a target set of desired cardinality consisting of influential nodes to maximize the volume of the information cascade.
Motivating Example 2: Weblogs

- In the domain of weblogs, bloggers publish posts and use hyper-links to refer to other posts and content on the web.
- Possible to observe the spread of information in the blogosphere, as each post is time stamped.
- In this setting, our goal is to select a small set of blogs (to read) which link to most of the stories that propagate over the blogosphere.
Models for Diffusion of Information

- Linear Thresholds Model
- Independent cascade model,
Models for Diffusion of Information

- Linear threshold model
- Independent cascade model,
Linear Thresholds Model

- Call a node active if it has adopted the information.
- Initially every node is inactive.
- Let us consider a node \( i \) and represent its neighbors by the set \( N(i) \).
- Node \( i \) is influenced by a neighbor node \( j \) according to a weight \( w_{ij} \). These weights are normalized in such a way that

\[
\sum_{j \in N(i)} w_{ij} \leq 1.
\]

- Further each node \( i \) chooses a threshold, say \( \theta_i \), uniformly at random from the interval \([0,1]\).
- This threshold represents the weighted fraction of node \( i \)'s neighbors that must become active in order for node \( i \) to become active.
Given a random choice of thresholds and an initial set (call it $S$) of active nodes, the diffusion process propagates as follows:

- in time step $t$, all nodes that were active in step $(t - 1)$ remain active
- we activate every node $i$ for which the total weight of its active neighbors is at least $\theta_i$
- if $A(i)$ is assumed to be the set of active neighbors of node $i$, then $i$ gets activated if
  \[ \sum_{j \in A(i)} w_{ij} \geq \theta_i. \]
- This process stops when there is no new active node in a particular time interval
Linear Thresholds Model: An Example

θ = 0.64
Linear Thresholds Model: An Example

\[ \theta = 0.64 \]
Linear Thresholds Model: An Example
Linear Thresholds Model: An Example

\(0.41 + 0.25 > \theta(=0.64)\)

\(\theta = 0.64\)

Active Node
Linear Thresholds Model: An Example

\[ 0.41 + 0.25 > \theta(=0.64) \]

- \[ \theta \] (Active Node)
Top-\(k\) Nodes Problem

- **Objective function** \((\sigma(.))\): Expected number of active nodes at the end of the diffusion process
- If \(S\) is the initial set of target nodes, then \(\sigma(S)\) is the expected number of active nodes at the end of the diffusion process
- For economic reasons, we want to limit the size of \(S\)
- For a given constant \(k\), the top-\(k\) nodes problem seeks to find a subset of nodes \(S\) of cardinality \(k\) that maximizes the expected value of \(\sigma(S)\)
Discovering Influential Individuals for Viral Marketing

Applications

- Databases
- Water Distribution Networks
- Blogspace
- Newsgroups
- Virus propagation networks


A Glimpse of State-of-the-Art

  - Introduced this problem as an algorithmic problem
  - A model using Markov Random Fields
  - Show that selecting the right set of users for a marketing campaign can make a substantial difference.

  - Show that the optimization problem of selecting most influential nodes is NP-hard problem.
  - Show that this objective function is a sub-modular function.
  - Propose a greedy algorithm that achieves an approximation guarantee of \((1 - \frac{1}{e})\).

1. Set $A \leftarrow \emptyset$.
2. for $i = 1$ to $k$ do
3. Choose a node $n_i \in N \setminus A$ maximizing $\sigma(A \cup \{n_i\})$
4. Set $A \leftarrow A \cup \{n_i\}$.
5. end for

- Running time of Greedy Algorithm: $O(knRm)$. 


- Develop an efficient algorithm that is reportedly 700 times faster than the greedy algorithm (KKT (2003)).
- There are two aspects to this speed up:
  1. Speeding up function evaluations using the sparsity of the underlying problem, and
  2. Reducing the number of function evaluations using the submodularity of the influence functions.
A Glimpse of State-of-the-Art (Cont.)

Research Gaps

- All the existing approximation algorithms are sensitive to the value of $k$.
- All the approximation algorithms crucially depend on the submodularity of the objective function. It is quite possible that the objective function can be non-submodular.
Our Proposed Approach

- We present a cooperative game theoretic framework for the top-$k$ nodes problem.
- We measure the influential capabilities of the nodes as provided by the Shapley value.
- ShaPley value based discovery of Influential Nodes (SPIN):
  1. Ranking the nodes,
  2. Choosing the top-$k$ nodes from the ranking order.
- Advantages of SPIN:
  1. Quality of solution is same as that of popular benchmark approximation algorithms,
  2. Works well for both sub-modular and non-submodular objective functions,
  3. Running time is independent of the value of $k$. 
Ranklist Construction

Let $\pi_j$ be the $j$-th permutation in $\hat{\Omega}$ and $R$ be repetitions.

Set $MC[i] \leftarrow 0$, for $i = 1, 2, \ldots, n$.

for $j = 1$ to $t$ do

Set $temp[i] \leftarrow 0$, for $i = 1, 2, \ldots, n$.

for $r = 1$ to $R$, do

assign random thresholds to nodes;

for $i = 1$ to $n$, do

$temp[i] \leftarrow temp[i] + v(S_i(\pi_j \cup \{i\}) - v(S_i(\pi_j))$

for $i = 1$ to $n$, do

$MC[i] \leftarrow temp[i]/R$;

for $i = 1$ to $n$, do

compute $\Phi[i] \leftarrow \frac{MC[i]}{t}$

Sort nodes based on the average marginal contributions of the nodes.
Initially all nodes are inactive.

Randomly assign a threshold to each node.

Fix a permutation $\pi$ and activate $\pi(1)$ to determine its influence.

Next consider $\pi(2)$. If $\pi(2)$ is already activated, then the influence of $\pi(2)$ is 0. Otherwise, activate $\pi(2)$ to determine its influence.

Continue up to $\pi(n)$.

Repeat the above process $R$ times (for example 10000 times) using the same $\pi$.

Repeat the above process $\forall \pi \in \hat{\Omega}$.
Choosing Top-$k$ Nodes

1. Naive approach is to choose the first $k$ in the RankList as the top-$k$ nodes.

2. *Drawback*: Nodes may be clustered.

3. RankList = \{5, 4, 2, 7, 11, 15, 9, 13, 12, 10, 6, 14, 3, 1, 8\}.

4. Top 4 nodes, namely \{5, 4, 2, 7\}, are clustered.

5. Choose nodes:
   - rank order of the nodes
   - spread over the network
<table>
<thead>
<tr>
<th>$k$ value</th>
<th>Greedy Algorithm</th>
<th>Shapley Value Algorithm</th>
<th>MDH based Algorithm</th>
<th>HCH</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>4</td>
<td>4</td>
<td>4</td>
<td>2</td>
</tr>
<tr>
<td>2</td>
<td>8</td>
<td>7</td>
<td>7</td>
<td>4</td>
</tr>
<tr>
<td>3</td>
<td>10</td>
<td>10</td>
<td>8</td>
<td>6</td>
</tr>
<tr>
<td>4</td>
<td>12</td>
<td>12</td>
<td>8</td>
<td>7</td>
</tr>
<tr>
<td>5</td>
<td>13</td>
<td>13</td>
<td>10</td>
<td>8</td>
</tr>
<tr>
<td>6</td>
<td>14</td>
<td>14</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>7</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>8</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>8</td>
</tr>
<tr>
<td>9</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>10</td>
</tr>
<tr>
<td>10</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>11</td>
</tr>
<tr>
<td>11</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>15</td>
<td>13</td>
<td>13</td>
</tr>
<tr>
<td>13</td>
<td>15</td>
<td>15</td>
<td>14</td>
<td>14</td>
</tr>
<tr>
<td>14</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
<tr>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
<td>15</td>
</tr>
</tbody>
</table>
Running Time of SPIN

- Overall running time of SPIN is $O(t(n + m)R + n \log(n) + kn + kRm)$ where $t$ is a polynomial in $n$.

- For all practical graphs (or real world graphs), it is reasonable to assume that $n < m$. With this, the overall running time of the SPIN is $O(tmR)$ where $t$ is a polynomial in $n$. 
Experimental Results: Data Sets

- **Random Graphs**
  - Sparse Random Graphs
  - Scale-free Networks (Preferential Attachment Model)

- **Real World Graphs**
  - Co-authorship networks,
  - Networks about co-purchasing patterns,
  - Friendship networks, etc.
## Experimental Results: Data Sets

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Number of Nodes</th>
<th>Number of Edges</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sparse Random Graph</td>
<td>500</td>
<td>5000 (approx.)</td>
</tr>
<tr>
<td>Scale-free Graph</td>
<td>500</td>
<td>1250 (approx.)</td>
</tr>
<tr>
<td>Political Books</td>
<td>105</td>
<td>441</td>
</tr>
<tr>
<td>Jazz</td>
<td>198</td>
<td>2742</td>
</tr>
<tr>
<td>C elegans</td>
<td>306</td>
<td>2345</td>
</tr>
<tr>
<td>NIPS</td>
<td>1061</td>
<td>4160</td>
</tr>
<tr>
<td>Netscience</td>
<td>1589</td>
<td>2742</td>
</tr>
<tr>
<td>HEP</td>
<td>10748</td>
<td>52992</td>
</tr>
</tbody>
</table>
Experiments: Random Graphs

- **Greedy Algorithm**
- **SPIN Algorithm**
- **MDH**
- **HCH**
Experiments: Real World Graphs

Graph 1: Number of Active Nodes vs. Size of Initial Target Set for Greedy Algorithm, SPIN Algorithm, MDH, and HCH.

Graph 2: Number of Active Nodes vs. Size of Initial Target Set for Greedy Algorithm, SPIN Algorithm, MDH, and HCH.

Graph 3: Number of Active Nodes vs. Size of Initial Target Set for Greedy Algorithm, SPIN Algorithm, MDH, and HCH.

Graph 4: Number of Active Nodes vs. Initial Target Set Size for Greedy Algorithm and SPIN Algorithm.
Top-10 Nodes in Jazz Dataset
Top-10 Nodes in NIPS Co-Authorship Data Set
## Running Times: SPIN vs KKT

<table>
<thead>
<tr>
<th>Dataset</th>
<th>Nodes</th>
<th>SPIN (MIN)</th>
<th>KKT (MIN)</th>
<th>Speed-up</th>
</tr>
</thead>
<tbody>
<tr>
<td>Random graph ( (p = 0.005) )</td>
<td>500</td>
<td>13.9</td>
<td>824.93</td>
<td>59</td>
</tr>
<tr>
<td>Random graph ( (p = 0.01) )</td>
<td>500</td>
<td>14.8</td>
<td>1123.16</td>
<td>75</td>
</tr>
<tr>
<td>Random graph ( (p = 0.02) )</td>
<td>500</td>
<td>16.3</td>
<td>1302.46</td>
<td>79</td>
</tr>
<tr>
<td>Political Books</td>
<td>105</td>
<td>0.89</td>
<td>44.64</td>
<td>50</td>
</tr>
<tr>
<td>Jazz</td>
<td>198</td>
<td>1.1</td>
<td>366</td>
<td>332</td>
</tr>
<tr>
<td>C.elegans</td>
<td>306</td>
<td>14.02</td>
<td>901</td>
<td>64</td>
</tr>
<tr>
<td>NIPS</td>
<td>1062</td>
<td>15.2</td>
<td>7201.54</td>
<td>473</td>
</tr>
<tr>
<td>Network-Science</td>
<td>1589</td>
<td>28.25</td>
<td>8539.48</td>
<td>302</td>
</tr>
</tbody>
</table>

**Table:** Speedup of the SPIN algorithm to find Top-30 nodes on various datasets compared to that of KKT algorithm
### Running Times: SPIN vs KKT

<table>
<thead>
<tr>
<th>Top-(k) Nodes</th>
<th>Running Time (in MINUTES)</th>
<th>Speed-up of SPIN over KKT</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>SPIN Algorithm</td>
<td>KKT Algorithm</td>
</tr>
<tr>
<td>(k = 10)</td>
<td>28.04</td>
<td>1341.29</td>
</tr>
<tr>
<td>(k = 20)</td>
<td>28.09</td>
<td>4297.02</td>
</tr>
<tr>
<td>(k = 30)</td>
<td>28.13</td>
<td>8539.48</td>
</tr>
<tr>
<td>(k = 40)</td>
<td>28.18</td>
<td>13949.9</td>
</tr>
<tr>
<td>(k = 50)</td>
<td>28.25</td>
<td>20411.1</td>
</tr>
</tbody>
</table>

**Table:** Running times of the SPIN, KKT, and LKG algorithms on the NetScience data set \(n = 1589\) to determine top-\(k\) nodes where \(k = 10, 20, 30, 40, 50\) and the speed up of the SPIN algorithm over the KKT algorithm.
Running Times: SPIN vs LKG on Random Graphs

![Graph showing running times for SPIN and LKG algorithms on random graphs. The x-axis represents different data sets (RG (p=0.005), RG (p=0.01), RG (p=0.02)), and the y-axis represents running time to determine top-30 nodes (in minutes). The graph compares the performance of SPIN and LKG algorithms across these data sets.]
Running Times: SPIN vs LKG on Real World Graphs
Minimum Thresholds Model

- Each node $i$ initially chooses a threshold $\theta_i$ uniformly at random from the interval $[0, 1]$,
- Node $i$ becomes active in time step $t$ if $f_i(S) \geq \theta_i$, where $S \subseteq N_i$ is the set of active neighbors of $i$ in step $(t - 1)$,
- We define the threshold function $f_i$ as follows:

$$f_i(S) = \min_{j \in S} \{ \alpha_j w_{ij} \}$$  \hspace{1cm} (1)

where $\alpha_j \geq 0$, $\forall j \in N_i$.

**Lemma:** Given the minimum linear threshold model and for any node $i$, the threshold function $f_i$ is monotone decreasing and supermodular.
Experiments: Minimum Thresholds Model

(i) SPIN Algorithm
   KKT Algorithm

(ii) SPIN Algorithm
     KKT Algorithm

Number of Active Nodes vs Initial Target Set Size
Multiplication Thresholds Model

- Each node $i$ initially chooses a threshold $\theta_i$ uniformly at random from the interval $[0, 1]$.
- Node $i$ becomes active in time step $t$ if $f_i(S) \geq \theta_i$, where $S \subseteq N_i$ is the set of active neighbors of $i$ in step $(t - 1)$.
- We define the threshold function $f_i$ as follows:

$$f_i(S) = \prod_{j \in S} w_{ij} \quad (2)$$

where $w_{ij}$ is the normalized weight representing the level of influence of node $j$ on node $i$ such that $\sum_{j \in N_i} w_{ij} \leq 1$.

**Lemma:** Given the multiplication threshold model and any node $i \in N$, the threshold function $f_i$ is monotone decreasing and supermodular.
Discovering Influential Individuals for Viral Marketing

Experiments: Multiplication Thresholds Model

(i) SPIN Algorithm
(ii) SPIN Algorithm
(iii) SPIN Algorithm
(iv) SPIN Algorithm

Graphs showing the number of active nodes against the size of the initial target set for different algorithms.
Next Part of the Talk

1. Social Network Analysis: Quick Primer
2. Foundational Concepts in Game Theory
3. Discovering Influential Individuals for Viral Marketing
4. Social Network Formation
5. Community Detection in Social Networks
6. Query Incentive Networks
7. Summary and To Probe Further
Graph Partitioning

Given an undirected and unweighted graph $G$, partition the nodes of $G$ into disjoint groups (or communities), such that the nodes in each group are densely connected themselves.

The partition of a graph is a set $\Pi = \{S_k\}_{k=1}^K$ that partitions the set of nodes $N$ of the graph into $K$ groups such that

1. $S_k \subseteq N$ for $k = 1, 2, \ldots, K$,
2. $S_x \cap S_y = \emptyset$ for $x, y = 1, 2, \ldots, K$,
3. $\bigcup_{k=1}^K S_k = N$.
4. Each $S_k$ is dense.

We call each $S_k$ a group or a community.
Community Detection in Social Networks

Graph Partitioning: An Example

\[ \Pi = \{S_1, S_2\} \text{ where } S_1 = \{1, 2, 3, 4, 5, 6\} \text{ and } S_2 = \{7, 8, 9, 10, 11\} \]
Graph Partitioning: Applications

1. VLSI circuit design
2. Parallel computing
3. Social network analysis
4. Graph visualization and summarization

Graph Visualization and Summarization
Graph Visualization and Summarization
Graph Visualization and Summarization
Approaches for Graph Partitioning

- Spectral Approach
- Geometric Approach
- Multi-level Approach
- Social Network Analysis Approach
  - Algorithms based on Centrality Measures
  - Algorithms based on Random Walks
  - Algorithms based on Optimization

Research Gap and Motivation

First, many of these algorithms work with a global objective function (such as modularity, conductance) for detecting communities in networks.

Communities in social networks emerge due to the actions of autonomous individuals in the network, without regard to a central authority enforcing certain global objective function.

Thus, algorithms that work with global objective functions do not satisfactorily explain the emergence of communities in social networks.

Second, many algorithms for community detection require the number of communities to be detected as input.
Community Detection Problem

- **Problem Statement**: Given an undirected and unweighted graph, we want to design a decentralized algorithm (due to autonomous actions of individuals in the network) that determines a partitioning of the graph with appropriate number of groups (or communities) such that each group in that partition is **dense**

- We propose a game theoretic approach to address the problem

**Perhaps first game theoretic approach for community detection**
Challenges Involved (Also Wish List)

- Algorithm should use only local information
- Existence of an equilibrium to be guaranteed
- Communities should be dense at equilibrium
- Algorithm itself should determine the number of communities
Community Detection Problem: Our Contributions

- Game theoretic framework based on Nash stable partition
- Propose a utility function based on only local information - it guarantees the existence of a Nash stable partition
- Lower bound on the coverage of any Nash stable partition
- Equivalence of NSP with another popular notion of equilibrium partition under the proposed utility function
- Leads to an efficient algorithm
  - SCoDA (Stable COmmunity Detection Algorithm)
- SCoDA does not require the number of communities as input
Graph Partitioning: Basic Definitions

- Let $G = (N, E)$ be an undirected and unweighted graph where $N = \{1, 2, \ldots, n\}$ is the finite set of nodes and $E$ is the set of edges.
- $N_i$ is the set of all neighbors of $i$ including $i$.
- $A_i = \{H \mid H \text{ is a subgraph of } G \text{ and } i \in H\}$
- $(N, u_i)$ is the graph partitioning game where $N$ is the set of nodes in $G$ and $u_i : A_i \rightarrow \mathbb{R}$ is the utility of node $i$.
- $\Psi$ is the set of all partitions.
- $\Pi(i) = \{S \mid S \in \Pi \text{ and } S \cap N_i \neq 0\}$
Graph Partitioning: Basic Definitions

Nash Stable Partition (NSP): A partition $\Pi \in \Psi$ is called a Nash stable partition of the given graph if $\forall i \in N$,

$$u_i(S_{\Pi}(i), G) \geq u_i(S_k \cup \{i\}, G), \forall S_k \in \Pi(i).$$

Individually Stable Partition (ISP): A partition $\Pi \in \Psi$ is called an Individually stable partition of the given graph if there does not exist $i \in N$ and a group $S_k \in \Pi$ such that

$$u_i(S_k \cup \{i\}) > u_i(S_{\Pi}(i)), \text{ and } u_j(S_k \cup \{i\}) > u_j(S_k), \forall j \in S_k.$$

Proposed Utility Function

- Utility function should use only local information
- It should ensure the existence of NSP
- It should produce dense communities at NSP
- For each $i \in N$, the utility of node $i$ is defined to be the number of neighbors of that node in its community plus a function of the fraction of neighbors in its community that are connected themselves.
Proposed Utility Function (Cont.)

- For each $i \in N$ and $\forall S \in A_i$,

$$u_i(S) = d_i(S) + \frac{T_i(S)}{d_i(S)} f(d_i(S))$$

where (i) $d_i(S)$ is the number of neighbors of node $i$ in community $S$, and (ii) $T_i(S)$ is number of pairs of neighbors of node $i$ in $S$ that are connected themselves, and (iii) $f(d_i(S))$ is a weight function

- To keep matters simple, we consider linear weight function $f(.)$ such as:

$$f(d_i(S)) = \alpha d_i(S), \, \forall i \in N, \, \forall S \in A_i$$

where $\alpha > 0$ is a constant

- To keep matters further simple, we work with $\alpha = 1$ and $\alpha = 2$
Proposed Utility Function (Cont.)

- When $\alpha = 1$, for each $i \in N$ and $\forall S \in A_i$,

$$u_i(S) = d_i(S) + \frac{2 T_i(S)}{d_i(S) - 1}$$

- When $\alpha = 2$, for each $i \in N$ and $\forall S \in A_i$,

$$u_i(S) = d_i(S) + \frac{4 T_i(S)}{d_i(S) - 1}$$
Proposed Utility Function (Cont.)

**Lemma:** When $\alpha = 1$, it holds that $u_i(S) \leq u_i(\bar{S})$ for each $S \subseteq \bar{S}$.

- This can be proved using simple algebra
- The utility of node $i$ does not decrease when new nodes join its community
- Suitable for certain practical applications such as collaborative editing in Wikipedia

**Lemma:** When $\alpha = 2$, for each $S \subseteq \bar{S}$, then $u_i(S) \leq u_i(\bar{S})$ holds only if $T_i(S) \leq \frac{d_i(S)(d_i(S)-1)}{4}$, $\forall i \in S$

- The utility of node $i$ need not necessarily increase when new nodes join its community
- Suitable for certain practical settings such as collaboration networks
Nash Stable Partition: An Example

\[ \Pi = \{ S_1, S_2 \} \] where

\[ S_1 = \{1, 2, 3, 4, 5, 6\}, \quad \text{and} \quad S_2 = \{7, 8, 9, 10, 11\} \]

\[ u_1(S_1) = 4, \quad u_1(S_2) = 0; \quad u_7(S_2) = 6, \quad u_7(S_1) = 1; \]
\[ u_2(S_1) = 6.668, \quad u_2(S_2) = 0; \quad u_8(S_2) = 6, \quad u_8(S_1) = 0; \]
\[ u_3(S_1) = 6.66, \quad u_3(S_2) = 0; \quad u_9(S_2) = 6.66, \quad u_9(S_1) = 0; \]
\[ u_4(S_1) = 6, \quad u_4(S_2) = 0; \quad u_{10}(S_2) = 6.66, \quad u_{10}(S_1) = 0; \]
\[ u_5(S_1) = 6, \quad u_5(S_2) = 1; \quad u_{11}(S_2) = 4, \quad u_{11}(S_1) = 0; \]
\[ u_6(S_1) = 1, \quad u_6(S_2) = 1; \]
Important Properties of NSPs

For each $i \in N$, let $\Psi(\Pi, i)$ be the set of all partitions where each partition is derived from $\Pi$ by moving node $i$ from $S_{\Pi}(i)$ to some $X \in \Pi(i)$. That is, $|\Psi(\Pi, i)| = |\Pi(i)| - 1$, (because $S_{\Pi}(i) \in \Pi(i)$).

Let $INTRA(\Pi) = \{(i, j) \in E \mid \exists S \in \Pi \ni i, j \in S\}$ be the set of edges within communities in a partition $\Pi$. We also define $INTER(\Pi) = E \setminus INTRA(\Pi)$ to be the set of edges across communities in partition $\Pi$.

Consider $i, j, k \in N$. If $(i, j) \in E$, $(j, k) \in E$, and $(k, i) \in E$, then we say that nodes $i, j, k$ form a triangle in the graph.

Finally, let $E(S)$ be the set of all edges among nodes in $S$ only. That is, $E(S) = \{(i, j) \in E \mid i, j \in S\}$. 

Lemma: Assume that $\alpha = 1$. Given an undirected and unweighted graph $G = (N, E)$ and any Nash stable partition $\Pi = \{S_1, S_2\}$ with two communities of $G$, then $\text{coverage}(\Pi) \geq \frac{1}{3}$.

Proof Sketch:

- For each $i \in S_1$, we have that
  
  $$d_i(S_2) + \frac{2T_i(S_2)}{d_i(S_2) - 1} \leq d_i(S_1) + \frac{2T_i(S_1)}{d_i(S_1) - 1} \tag{3}$$

- Now using fact that $|\text{INTER}(\Pi)| = \sum_{i \in S_1} d_i(S_2)$, we get that
  
  $$|\text{INTER}(\Pi)| + \sum_{i \in S_1} \frac{2T_i(S_2)}{d_i(S_2) - 1} \leq \sum_{i \in S_1} d_i(S_1) + \sum_{i \in S_1} \frac{2T_i(S_1)}{d_i(S_1) - 1}. \tag{4}$$
Important Properties of NSPs (Cont.)

- On similar lines, repeating the above argument for $S_2$, we get that

$$|\text{INTER}(\Pi)| + \sum_{i \in S_2} \frac{2T_i(S_1)}{d_i(S_1) - 1} \leq \sum_{i \in S_2} d_i(S_2) + \sum_{i \in S_2} \frac{2T_i(S_2)}{d_i(S_2) - 1}$$

(5)

- After summing the expressions (4) and (5), we use the fact that

$\sum_{i \in S_1} d_i(S_1) = 2|E(S_1)|$, $\sum_{i \in S_2} d_i(S_2) = 2|E(S_2)|$, and rearranging terms, we get that

$$\sum_{i \in S_1} \frac{T_i(S_2)}{d_i(S_2) - 1} + \sum_{i \in S_2} \frac{T_i(S_1)}{d_i(S_1) - 1} + |\text{INTER}(\Pi)| \leq$$

$$|E(S_1)| + |E(S_2)| + \sum_{i \in S_1} \frac{T_i(S_1)}{d_i(S_1) - 1} + \sum_{i \in S_2} \frac{T_i(S_2)}{d_i(S_2) - 1}$$
Important Properties of NSPs (Cont.)

- Now substituting $INTER(\Pi) = E \setminus INTRA(\Pi)$ and $E(S_1) \cup E(S_2) = INTRA(\Pi)$ in the above expression and readjusting the terms, we get that

\[ \Rightarrow |INTRA(\Pi)| \geq \frac{1}{2} |E| - \frac{1}{2} \left[ \sum_{i \in S_1} \frac{T_i(S_1)}{d_i(S_1)} - 1 + \sum_{i \in S_2} \frac{T_i(S_2)}{d_i(S_2)} - 1 \right] \]

- Since the maximum value for $T_i(X)$ is $\frac{d_i(X)(d_i(X)-1)}{2}$ for each $X \in \{S_1, S_2\}$; and bounding the above expression using this fact, we get that

\[ |INTRA(\Pi)| \geq \frac{1}{2} |E| - \frac{1}{2} \left[ \sum_{i \in S_1} \frac{d_i(S_1)}{2} + \sum_{i \in S_2} \frac{d_i(S_2)}{2} \right] \]

\[ \Rightarrow |INTRA(\Pi)| \geq \frac{1}{2} |E| - \frac{1}{2} |INTRA(\Pi)| \quad \Rightarrow \text{coverage}(\Pi) \geq \frac{1}{3}. \]
Lemma: Assume that $\alpha = 2$. Given an undirected and unweighted graph $G = (N, E)$ and a Nash stable partition $\Pi = \{S_1, S_2\}$ with 2 communities of $G$, then $\text{coverage}(\Pi) \geq \frac{1}{4}$. 
An Algorithm for Graph Partitioning

- SCoDA: Stable COmmunity Detection Algorithm
  - Initial configuration
  - Order of nodes
  - Stopping criterion
- Running time: $O(n \log(n) + nmd_{\text{max}}^2)$
- The resultant community structure with SCoDA is obviously a Nash stable partition
1: Let $\Pi$ be the initial partition of the graph $G$.
2: Let $visit\_order[]$ contains nodes in non-decreasing order of the degree.
3: while true, do
4: \hspace{1em} for $i := 1$ to $n$
5: \hspace{2em} flag $\leftarrow 0$;
6: \hspace{2em} $j \leftarrow visit\_order[i]$
7: \hspace{2em} if $u_i(S_{\Pi}(i)) < u_i(S_k)$ for some $S_k \in \Pi(i)$, then
8: \hspace{3em} move node $i$ from $S_{\Pi}(i)$ to $S_k$;
9: \hspace{3em} flag $\leftarrow 1$;
10: \hspace{2em} end if
11: \hspace{1em} end for
12: \hspace{1em} if flag $= 0$, then
13: \hspace{2em} break;
14: \hspace{1em} end if
15: end while
**Lemma:** When $\alpha = 1$, Algorithm 1 always guarantees convergence to a Nash stable partition, given any undirected and unweighted graph.

- Given that $\alpha = 1$. We first define a function $\Phi : \Psi \rightarrow \mathbb{N}$ such that, for each $\Pi \in \Psi$, $\Phi(\Pi)$ represents the sum of the number of intra-community edges in $\Pi$ and the number of triangles within the communities in $\Pi$.

- For each $\Pi \in \Psi$, we call $\Phi(\Pi)$ as the **capacity** of the partition $\Pi$. More formally, $\forall \Pi \in \Psi$,

$$
\Phi(\Pi) = \sum_{S \in \Pi} \sum_{j \in S} \frac{d_j(S)}{2} + \sum_{S \in \Pi} \sum_{i \in S} \sum_{p, q \in S \cap N_i} \frac{I(p, q)}{3}
$$

(6)

where $I(p, q)$ is an indicator function that takes value 1 if nodes $p$ and $q$ are adjacent in $G$ and 0 otherwise.
Existence of NSP (Cont.)

- Define a partition \( \Pi_x \in \Psi \) to be *maximal* if there does not exist any \( \Pi_y \in \Psi \) such that (i) \( \Pi_y \in \Psi(\Pi_x, i) \) for some \( i \in \mathcal{N} \), and (ii) the capacity of \( \Pi_y \) is strictly greater than the capacity of \( \Pi_x \).

- Now consider a partition \( \Pi_1 \in \Psi \). Let \( \Pi_2 \in \Psi(\Pi_1, i) \) be a partition that is obtained from \( \Pi_1 \), when node \( i \) jumps from \( S_{\Pi_1}(i) \) to some \( X \in \Pi_1(i) \).

- Note that node \( i \) jumps from \( S_{\Pi_1}(i) \) to some \( X \in \Pi_1(i) \) to improve its utility, then

\[
d_i(S_{\Pi_1}(i)) + \frac{2T_i(S_{\Pi_1}(i))}{d_i(S_{\Pi_1}(i))} - 1 < d_i(X) + \frac{2T_i(X)}{d_i(X)} - 1.
\]  

(7)
Existence of NSP (Cont.)

- This is possible when either $d_i(X) > d_i(S_{\Pi_1}(i))$ or $T_i(X) > T_i(S_{\Pi_1}(i))$ holds.
- A simple algebra based on this fact implies that $\Phi(\Pi_2) > \Phi(\Pi_1)$.
- This further implies that, whenever a node moves from one group to the other group, the capacity of the new partition strictly improves upon the capacity of the old partition.
- Since the number of partitions is finite, a maximal partition certainly exists.
Improved Version of SCoDA

- We further refine the Nash stable partition produced through SCoDA to improve its modularity.

- **Greedy Strategy:** In each step, we determine a pair of communities to merge so that the modularity of the resultant community structure is maximized. We repeat this step until we do not find any pair of communities to merge to improve modularity.

- Running time of the improved version of SCoDA is $O(n \log(n) + nmd_{\text{max}}^2 + k^3 n)$. 


Benchmark Algorithms for Comparison

1. Edge Betweenness Algorithm (Girvan and Newman (2002)),
2. Fast Greedy Algorithm (Newman (2004)),
3. Spectral Algorithm (Newman (2006)).
4. A Randomized and Game Theoretic Algorithm (Chen et. al. (2010))
Modularity of a partition $\Pi$ of given undirected and unweighted graph $G = (\mathcal{N}, \mathcal{E})$ is:

$$Q(\Pi, G) = \frac{1}{2m} \sum_{i,j \in \mathcal{N}} \left( a_{i,j} - \frac{n_i n_j}{2m} \right) \delta(S_{\Pi}(i), S_{\Pi}(j))$$

where (i) $m$ is number of edges in $G$, (ii) $n_x$ is degree of node $x \in \mathcal{N}$, (iii) $S_{\Pi}(x)$ represents ID of the community of node $x$, and (iv) $\delta(a, b)$ takes value 1 if $a = b$ and 0 otherwise.

## Experimental Results: Data Sets

<table>
<thead>
<tr>
<th>Data Set</th>
<th>Nodes</th>
<th>Edges</th>
<th>Triangles</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>34</td>
<td>78</td>
<td>45</td>
</tr>
<tr>
<td>Dolphins</td>
<td>62</td>
<td>318</td>
<td>95</td>
</tr>
<tr>
<td>Les Miserables</td>
<td>77</td>
<td>508</td>
<td>467</td>
</tr>
<tr>
<td>Political Books</td>
<td>105</td>
<td>882</td>
<td>560</td>
</tr>
<tr>
<td>FootBall</td>
<td>115</td>
<td>1226</td>
<td>810</td>
</tr>
<tr>
<td>Jazz</td>
<td>198</td>
<td>2742</td>
<td>17899</td>
</tr>
<tr>
<td>Email</td>
<td>1133</td>
<td>5451</td>
<td>10687</td>
</tr>
<tr>
<td>Yeast</td>
<td>2361</td>
<td>6913</td>
<td>5999</td>
</tr>
</tbody>
</table>

**Table:** Description of various real world network data sets
Comparison with GN Algorithm: Dolphins Data Set

- Modularity using GN Algorithm: 0.519.
- Modularity using SCoDA: 0.526.
Comparison with GN Algorithm: Football Data Set

- Modularity using GN Algorithm: 0.598.
- Modularity using SCoDA: 0.6.
Comparison with GN Algorithm: Les Miserable Data Set

- Modularity using GN Algorithm: 0.538.
- Modularity using SCoDA: 0.538.
Experimental Results - Modularity

**Table:** Comparison of modularity due to SCoDA with that of four benchmark algorithms.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>SCoDA ($\alpha = 1$)</th>
<th>SCoDA ($\alpha = 2$)</th>
<th>GN</th>
<th>Greedy</th>
<th>Spectral</th>
<th>RGT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>0.4</td>
<td>0.4</td>
<td>0.4</td>
<td>0.38</td>
<td>0.393</td>
<td>0.392</td>
</tr>
<tr>
<td>Dolphins</td>
<td>0.525</td>
<td>0.525</td>
<td>0.519</td>
<td>0.495</td>
<td>0.491</td>
<td>0.502</td>
</tr>
<tr>
<td>Les-Mis</td>
<td>0.545</td>
<td>0.545</td>
<td>0.538</td>
<td>0.5</td>
<td>0.532</td>
<td>0.54</td>
</tr>
<tr>
<td>Pol-Books</td>
<td>0.524</td>
<td>0.524</td>
<td>0.496</td>
<td>0.509</td>
<td>0.469</td>
<td>0.493</td>
</tr>
<tr>
<td>Football</td>
<td>0.60</td>
<td>0.60</td>
<td>0.598</td>
<td>0.566</td>
<td>0.539</td>
<td>0.581</td>
</tr>
<tr>
<td>Jazz</td>
<td>0.439</td>
<td>0.439</td>
<td>0.403</td>
<td>0.438</td>
<td>0.393</td>
<td>0.439</td>
</tr>
<tr>
<td>Email</td>
<td>0.479</td>
<td>0.51</td>
<td>0.51</td>
<td>0.494</td>
<td>0.498</td>
<td>0.509</td>
</tr>
<tr>
<td>Yeast</td>
<td>0.571</td>
<td>0.571</td>
<td>0.568</td>
<td>0.571</td>
<td>0.497</td>
<td>-</td>
</tr>
</tbody>
</table>
Experimental Results - Coverage

**Table:** Comparison of coverage due to SCoDA with that of four benchmark algorithms.

<table>
<thead>
<tr>
<th>Data Set</th>
<th>SCoDA ($\alpha = 1$)</th>
<th>SCoDA ($\alpha = 2$)</th>
<th>GN</th>
<th>Greedy</th>
<th>Spectral</th>
<th>RGT</th>
</tr>
</thead>
<tbody>
<tr>
<td>Karate</td>
<td>82.05</td>
<td>82.05</td>
<td>75.64</td>
<td>30.76</td>
<td>25.24</td>
<td>68.52</td>
</tr>
<tr>
<td>Dolphins</td>
<td>80.50</td>
<td>80.50</td>
<td>82.38</td>
<td>22.01</td>
<td>22.64</td>
<td>69.43</td>
</tr>
<tr>
<td>Les-Mis</td>
<td>75.19</td>
<td>75.19</td>
<td>73.22</td>
<td>35.03</td>
<td>25.19</td>
<td>72.83</td>
</tr>
<tr>
<td>Pol-Books</td>
<td>89.11</td>
<td>89.11</td>
<td>91.83</td>
<td>59.63</td>
<td>45.57</td>
<td>74.45</td>
</tr>
<tr>
<td>Football</td>
<td>69.00</td>
<td>69.00</td>
<td>74.22</td>
<td>16.15</td>
<td>12.39</td>
<td>67.92</td>
</tr>
<tr>
<td>Jazz</td>
<td>79.83</td>
<td>79.83</td>
<td>77.93</td>
<td>31.72</td>
<td>34.31</td>
<td>71.35</td>
</tr>
<tr>
<td>Email</td>
<td>72.46</td>
<td>73.18</td>
<td>75.67</td>
<td>21.61</td>
<td>18.33</td>
<td>58.12</td>
</tr>
<tr>
<td>Yeast</td>
<td>66.99</td>
<td>67.85</td>
<td>74.55</td>
<td>19.53</td>
<td>22.14</td>
<td>-</td>
</tr>
</tbody>
</table>
In social networks, typically individuals belong to more than one community.

It is important to determine overlapping communities.

Community Detection in Social Networks

Community Formation Game

Let $G = (V, E)$ be an undirected and unweighted graph with $n = |V|$ and $m = |E|$ and it represents the underlying static acquaintance graph.

The set of all possible communities is denoted as $[k] = \{1, 2, \ldots, k\}$, where $k$ is polynomial in $n$.

Each strategy $L_i$ of $v_i$ is a subset of communities that it wants to join; i.e, $L_i \subseteq [k]$.

Let $L = (L_1, L_2, \ldots, L_n)$ be the profile of strategies of $n$ agents.

The utility of the $i$-th agent is measured by a gain function $g_i(L)$ and a loss function $l_i(L)$ and it is defined as

$$u_i(L) = g_i(L) - l_i(L)$$
In general, Nash equilibria may not exist in a community formation game. Consider an instance of community formation game where one node \( u \) always prefer to be with another node \( v \) in the same community while \( v \) always prefer not to be in the same community as \( u \).

A community formation game is a potential game if

\[
\Phi(L_{-i}, L'_i) - \Phi(L) = u_i(L_{-i}, L'_i) - u_i(L)
\]

for every strategy profile \( L \) and for every strategy \( L'_i \) of agent \( i \).
Community Detection in Social Networks

Community Formation Game (Cont.)

**Definition:** A set of functions \( \{f_i(.): 1 \leq i \leq n\} \) is locally linear with linear factor \( \rho \) if for every strategy profile \( L \) and every strategy \( L'_i \) of \( i \), the following relation holds: \( \forall i \in V, \)

\[
f_i(L_{-i}, L'_i) - f_i(L) = \rho(f_i(L_{-i}, L'_i) - f_i(L))
\]

**Theorem:** If \( \{g_i(.): i \in V\} \) and \( \{l_i(.): i \in V\} \) are locally linear functions with linear factor \( \rho_g \) and \( \rho_l \), then the community formation game is a potential game.

**Lemma:** There exists a community formation game, in which the sets of gain and loss functions are locally linear, such that both computing the best response for an individual agent and computing a Nash Equilibrium in the game are NP-hard.
Now we discuss a set of gain and loss functions and they can be computed efficiently.

**Gain Function:** We use here is a generalized version of modularity.
- Define $\delta(i, j) = 1$ if $|L_i \cap L_j| \geq 1$ and $\delta(i, j) = 0$ otherwise.
- Let $A$ be the adjacency matrix of graph $G$.
- $g_i(L) = \frac{1}{2m} \sum_{j \in [n]} \left( A_{ij} \delta(i, j) - \frac{d_i d_j}{2m} |L_i \cap L_j| \right)$

**Loss Function:** We use a simple loss function to model the aspect that an agent may suffer by joining new communities.
- Let $c > 0$ be a constant. Then, $l_i(L) = (|L_i| - 1)c$. 

Easy to verify that the above $g_i(L)$ and $l_i(L)$ are locally linear functions, with linear factor $\frac{1}{2}$ and 1 respectively.

**Theorem:** Given the above $g_i(L)$ and $l_i(L)$, the community formation game has a Nash equilibrium.
Computing the best response for each agent might be hard in certain contexts from the previous Lemma.

It is not reasonable to assume that individuals always make the best response.

An agent can only locally implement the following three operations:

- **Join**: Agent $i$ joins a new community on top of the communities she joins by adding a new label in $L_i$.
- **Leave**: Agent $i$ leaves a community she is in by removing a label from $L_i$.
- **Switch**: Agent $i$ switches from one community to another by replacing a label in $L_i$. 

Algorithm 1: LocalEquilibrium(G)

- initialize each node to a singleton community
- repeat the following process until no node can improve itself
  - randomly pick a node $i$ and perform the best operation among join, leave and switch

Entire strategy space of agent $i$ is $2^k$

For each agent $i$ with the current community label set $L_i$, we use $ls(L_i)$ to denote $i$'s local strategy space that is obtained by applying join, leave, or switch once on $L_i$
Specific Gain and Loss Functions (Cont.)

- **Local Equilibrium:** Given $G$, the strategy profile $L = (L_1, L_2, ..., L_n)$ forms a local equilibrium of the community formation game if all agents are playing their local optimal strategies, that is, $\forall i \in V$ and $L'_i \in ls(L_i)$,
  
  $$u_i(L_{-i}, L'_i) \leq u_i(L_{-i}, L_i)$$

- **Theorem:** Let $g_i(L)$ be the personalized modularity function and $l_i(L) = c(|L_i| - 1)$ be a linear loss function with constant $c$ satisfying $4cm^2$ is an integer. Algorithm 1 takes at most $O(m^2)$ steps to reach a local equilibrium.
Experimental Results

Zachary’s Karate Club Network:
Dolphin Network:
Next Part of the Talk

1. Social Network Analysis: Quick Primer
2. Foundational Concepts in Game Theory
3. Discovering Influential Individuals for Viral Marketing
4. Social Network Formation
5. Community Detection in Social Networks
6. Query Incentive Networks
7. **Summary and To Probe Further**
Summary of the Tutorial

- We first presented the important fundamental concepts in social network analysis and game theory.
- We then presented game theoretic models for four important problems in social network analysis:
  - Determining top-$k$ influential individuals
  - Social network formation
  - Community detection
  - Query incentive networks
- We also gave a systematic analysis of these game theoretic models to SNA.
To Probe Further: Important Research Directions

- **Altruistic Game Theoretic Models:**
  - Often individuals in social networks are not only rational and intelligent but also altruistic
  - Social network formation and Bargaining on social networks

- **Time Varying Graphs:** Typically, the structure of networks change over time. Designing game theoretic models for such time varying graphs is a challenging and interesting research direction

- **Probabilistic Graphs:**
  - Complex networks often entail uncertainty and thus can be modeled as probabilistic graphs
To Probe Further: Important Text Books


To Probe Further: Important References


- D. Dikshit and Y. Narahari. Truthful and Quality Conscious Query Incentive Networks. In Workshop on Internet and Network Economics (WINE), 2009.

To Probe Further: Useful Resources

**Network Data Sets:**

- MEJ Newman: http://www-personal.umich.edu/~mejn/netdata
- Albert L. Barabasi: http://www.nd.edu/~networks/resources.htm
- NIST Data Sets: http://math.nist.gov/~RPozo/complex_datasets.html
- ...
To Probe Further: Useful Resources (Cont.)


- Home page of Y. Narahari: http://lcm.csa.iisc.ernet.in/hari/

- Home page of Ramasuri Narayanam: http://lcm.csa.iisc.ernet.in/nrsuri/

- Blog on Social Networks: http://cs2socialnetworks.wordpress.com/
Thank You