Complex Network Theory
Social Network Theory - Community Structure

Lecture delivered by Prof. Niloy Ganguly
Scribed by Arnab Sinha
arnabsinha@gmail.com

March 24, 2006

1 Introduction

Community clustering is an important question for investigation in the theory of social sciences. We know that on one extreme, we have all the nodes disconnected, (after removal of all the edges) while, on the other hand, we have a fully connected component (it is a clique, if all possible edges are added to the graph). So, we need to define some metric to define the ideal cluster. We will try to address this particluar question in the following lecture - *Is there any metric to define ideal cluster in a network (more specific to social networks)?* In this lecture we will be discussing the following community clustering algorithms.

- Girvan Newman Algorithm
- Radicche's Algorithm
- Wu and Hubermum's Algorithm

2 Girvan Newman Algorithm

Let, in a network we begin with k clusters. Next, we would form the following matrix,

$$\begin{pmatrix} e_{11} & e_{12} & .. & e_{1k} \\ .. & .. & .. & .. \\ e_{k1} & e_{k2} & .. & e_{kk} \end{pmatrix}$$

where, e_{ij} is the number of edges between cluster i and cluster j and the diagonal element e_{ii} represents the number of edges between cluster i only. Ideally, non-diagonal elements should be zero (since in ideal cluster the components should be disconnected from each other). So deviation from this behavior can be a co-efficient for randomness of the network. We define the co-efficient as follows,

$$Q = \sum_{\forall i} (e_{ii} - a_i^2)^2$$

where, $a_i = \sum_{j=1}^{k} e_{ij}$. Also, the term $(e_{ii} - a_i^2)$ is denoted as the *affinity* of the i-th cluster. We can also normalize the co-efficient Q.

3 Radicche's Algorithm

The Radicche's algorithm is divisive in nature. It tries to find the *edge-custering coefficient* of each edge. Depending on that co-efficient, we remove the edge from the network. It finds the number of loops a particular edge is part of. Let, n_1 and n_2 be two nodes which are in the different clusters.

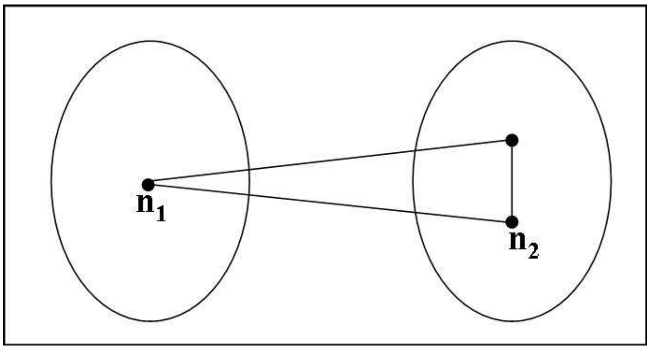

Figure 1: The Contribution of n_1

Suppose, that they form a loop of *three* members, then n_1 contributes 2 edges in between the clusters (since the third member is in either cluster of n_1 or that of n_2 and in any case, there would be two edges between the clusters). We define the edge-clustering co-efficient c_i,

$$c_i = \frac{z_i + 1}{min\{k_{n_1} - 1, k_{n_2} - 1\}}$$

where, z_i is the number of 3-edge loops that i-th node is part of and k_{n_i} is the degree of node n_i. We can remove that the edge with lowest value of c_i, which is actually acting like a bridge bwteen the two clusters.

4 Wu and Hubermum's Algorithm

In this algorithm, the concept of electrical circuits is brought in. We can map the network in the following manner. Simply replace the edges of the network with unit resistances (R=1 ohm) and add a voltage source as shown in Fig 2.

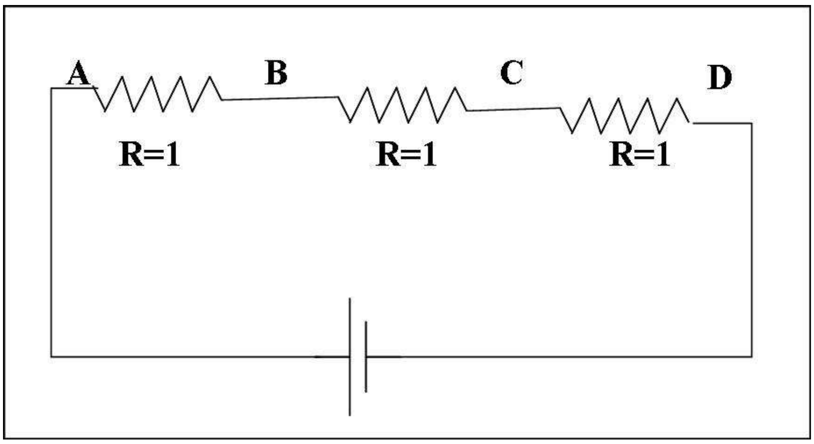

Figure 2: The Electrical Circuit

We can intuitively classify the nodes depending on the voltage at some point (say V_i). If $V_i \leq 0.5$, we infer that i-th node is in some cluster, while if $V_i \geq 0.5$ then i-th node belongs to some other cluster. From the *Kirchoff's Current Law*, we know that the net current at i-th node is nil. So, we get the following for the i-th node,

$$\sum_j I_{ij} = 0$$

$$\Rightarrow \quad \sum_j \frac{V_i - V_j}{R} = 0 \tag{1}$$

$$\Rightarrow \quad \sum_j (V_i - V_j) = 0, \text{ since, } R\text{=}1 \tag{2}$$

$$\Rightarrow \quad V_i = \frac{1}{k_i} \sum_j V_j, \; k_i \text{ is the no. of neighbors of } i\text{-th node} \tag{3}$$

$$\Rightarrow \quad V_i = \frac{1}{k_i} \sum_{j \text{ adjacent to } i} V_j \tag{4}$$

$$\Rightarrow \quad V_i = \frac{1}{k_i} \sum_{V(G)} V_j a_{ij} \tag{5}$$

$$\Rightarrow \quad V_i = \frac{1}{k_i} \sum_{V(G)-\{i,1\}} V_j a_{ij} + \frac{1}{k_i} a_{i1} \tag{6}$$

So, we can define another metric as follows,

$$A^{'} = \begin{pmatrix} \frac{a_{11}}{k_1} & .. & \frac{a_{1n}}{k_1} \\ .. & .. & .. \\ \frac{a_{n1}}{k_n} & .. & \frac{a_{nn}}{k_n} \end{pmatrix}$$

and,

$$A^{''} = \begin{pmatrix} \frac{a_{11}}{k_1} \\ .. \\ \frac{a_{n1}}{k_n} \end{pmatrix}$$

Now from eqn 6 we get the following.

$$\begin{aligned} & V = (A^{'}V + A^{''}) \\ \Rightarrow \;& V(1 - A^{'}) = A^{''} \\ \Rightarrow \;& V = A^{''}(1 - A^{'})^{-1} \end{aligned}$$