Communicability Graph and Community Structures in Complex Networks[1]

Alexander Vasilyev

University of Primorska.

May 30th. 2011

[1] E. Estrada, N. Hatano, Communicability graph and community structures in complex networks, Applied Mathematics and Computation 214 (2) (2009) 500 - 511.

Definitions.

In this talk we consider simple graphs $G = (V, E)$, having $|V| = n$ nodes and $|E| = m$ links, without self-loops or multiple links between nodes.

Definition

A *walk of length k* is a sequence of (not necessarily different) vertices $v_0, v1, \ldots, v_{k-1}, v_k$ such that $\forall i = 1, 2, \ldots, k$ $\exists$ a link from v_{i-1} to v_i

Definition

Let $\mathbf{A}(G) = \mathbf{A}$ be the adjacency matrix of the graph G. Then the *moment* $\mu_k(p, q) = (\mathbf{A}^k)_{pq}$ gives the number of walks of length k from node p to q

Communicability[2]

We define the *communicability* between a pair of nodes p and q as a weighted sum of the moments $\mu_k(p, q)$

$$
\begin{aligned}
G_{p,q} &= \sum_{k=0}^{\infty} \frac{\mu_k(p,q)}{k!} = \sum_{k=0}^{\infty} \frac{(\mathbf{A}^k)_{pq}}{k!} = \left(e^{\mathbf{A}}\right)_{pq} \\
&= \sum_{j=0}^{n} \phi_j(p)\phi_j(q)e^{\lambda_j}
\end{aligned}
\tag{1}
$$

where $\phi_j(p)$ is the p-th component of the j-th orthonormal eigenvector of the adjacency matrix $\mathbf{A}$, which is associated with the eigenvalue λ_j.

[2] E. Estrada, N. Hatano, Communicability in complex networks, Physical Review E 77 (2008) 036111.

Definition

We call the eigenvalues of the adjacency matrix in non-increasing order $\lambda_1 \geq \lambda_2 \geq \cdots \geq \lambda_n$, the *spectrum* of the graph.

The communicability can be decomposed into several terms as

$$
\begin{aligned}
G_{p,q} =& \left[\phi_1(p)\phi_1(q)e^{\lambda_1}\right] \\
&+ \left[\sum_{2\leq j\leq n} \phi_j^+(p)\phi_j^+(q)e^{\lambda_j} + \sum_{2\leq j\leq n} \phi_j^-(p)\phi_j^-(q)e^{\lambda_j}\right] \\
&+ \left[\sum_{2\leq j\leq n} \phi_j^+(p)\phi_j^-(q)e^{\lambda_j} + \sum_{2\leq j\leq n} \phi_j^-(p)\phi_j^+(q)e^{\lambda_j}\right]
\end{aligned} \tag{2}
$$

$$
\begin{aligned}
\Delta G_{p,q} &= \left[\sum_{2 \le j \le n} \phi_j^+(p)\phi_j^+(q)e^{\lambda_j} + \sum_{2 \le j \le n} \phi_j^-(p)\phi_j^-(q)e^{\lambda_j} \right] \\
&\quad + \left[\sum_{2 \le j \le n} \phi_j^+(p)\phi_j^-(q)e^{\lambda_j} + \sum_{2 \le j \le n} \phi_j^-(p)\phi_j^+(q)e^{\lambda_j} \right] \qquad (3) \\
&= \sum_{j}^{\text{intracluster}} \phi_j(p)\phi_j(q)e^{\lambda_j} - \left| \sum_{j}^{\text{intercluster}} \phi_j(p)\phi_j(q)e^{\lambda_j} \right|
\end{aligned}
$$

Definition

$C \subseteq V$ is a *community* of G if, and only if, $\Delta G > 0 \quad \forall (p,q) \in C$

Communicability graph.

$$\Theta(x) = \begin{cases} 1 & \text{if } x \geq 0, \\ 0 & \text{if } x < 0 \end{cases} \tag{4}$$

Let $\mathbf{\Delta}(G)$ be a matrix whose (p, q) entries are given by $\Delta G_{p,q}$.

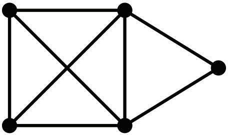
Definition

The *communicability graph* $\Theta(G)$ of the graph G is the graph whose adjacency matrix is given by $\Theta(\mathbf{\Delta}(G))$,
where $\Theta(\mathbf{\Delta}(G))$ results from the elementwise application of the function $\Theta(x)$ (4) to the matrix $\mathbf{\Delta}(G)$.

The nodes of the $\Theta(G)$ are the same as the nodes of G, and two nodes p and q in $\Theta(G)$ are connected if, and only if, $\Delta G_{p,q} > 0$.

Definition

A *clique* is a locally maximal complete subgraph.



Every community in the graph G is represented by a clique of accompanying communicability graph $\Theta(G)$.

Thus the problem of finding the communities of a graph G is transformed to the problem of finding the cliques of $\Theta(G)$.

All-clique problem.

That is, given a graph we need to determine all maximal complete subgraphs. This problem is a well-known NP-hard problem.

- Finding all cliques is expensive.
- The number of cliques can grow exponentially with every node added

Classic branch-and-bound approach for solving this problem is the Bron-Kerbosch algorithm[3] (1973).

- An algorithm to compute all cliques in linear time (relative to the number of cliques)
- Still widely used and referred to as one of the fastest algorithms.

[3] C. Bron, J. Kerbosch (1973), Algorithm 457: finding all cliques of an undirected graph, Commun. ACM (ACM) 16 (9): 575 - 577

Bron-Kerbosch algorithm

R - vertices of the current clique
P - pretenders that could be added to the clique
X - vertices used in previous steps
N_v - all the neighbors of v

Bron-Kerbosch recursive algorithm

BronKerbosch(R, P, X):
if $P = \emptyset$ and $X = \emptyset$: report R as a maximal clique
for each vertex $v \in P$:
BronKerbosch$\left(R \cup \{v\}, \quad P \cap N_v, \quad X \cap N_v\right)$
$P := P \setminus \{v\}$
$X := X \cup \{v\}$

Initial values:
$R = \emptyset, P = V, X = \emptyset$

Mergence of overlapping communities

Sorensen similarity index

$$S_{AB} = \frac{2|A \cap B|}{|A| + |B|} \qquad (5)$$

Sorensen index gives us an information how two communities are similar.
$0 \leq S_{AB} \leq 1$
We will merge two communities A and B if their index $S_{AB} \geq \alpha$

Let **S** be a matrix formed by Sorensen indeces for each pair of detected communities.

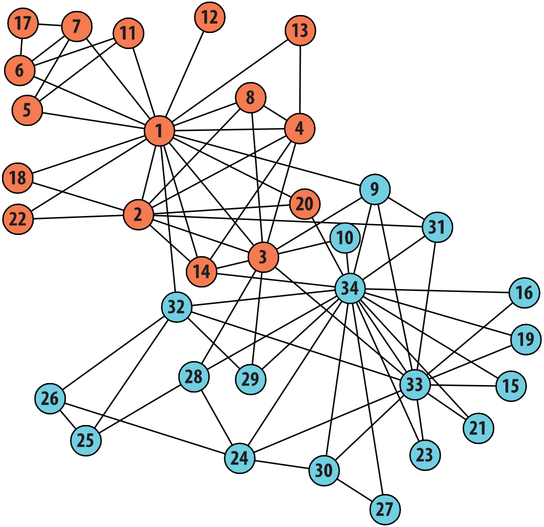
General procedure of managing overlapped communities

1. Find communities in the network using communicability graph conception;
2. Calculate S_{AB} for all pairs of communities and build the matrix $\mathbf{S}$;
3. For a given value of α, build the matrix $\mathbf{O}$, whose entries are

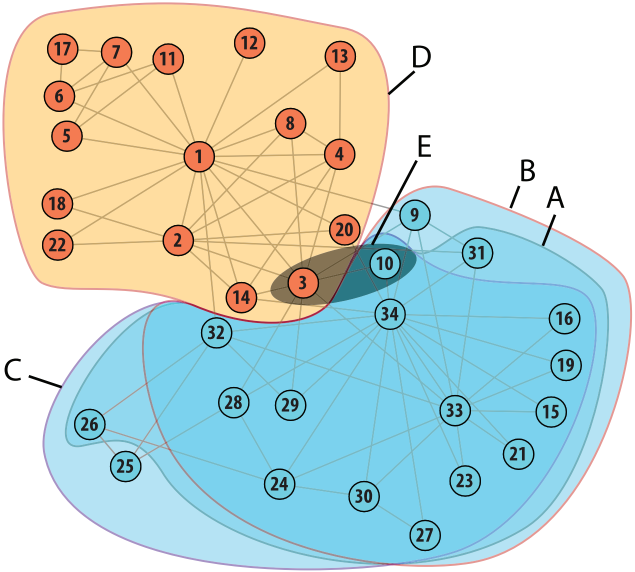
$$O_{AB} = \begin{cases} 1 & \text{if } S_{AB} \geq \alpha, \\ 0 & \text{if } S_{AB} < \alpha, \quad \text{or} \quad A = B \end{cases}$$

4. If $\mathbf{O} = \mathbf{0}$, STOP the process, else go to the next step;
5. Find cliques in the graph whose adjacency matrix is $\mathbf{O}$;
6. Merge the communities represented by the nodes of the cliques found on previous step and go to step 2.

Example: Zachary karate club



Example: Zachary karate club. Detected communities



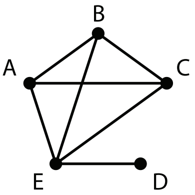
Example: Zachary karate club. Merging communities

Community-overlap matrix $\mathbf{S}$ for the network:

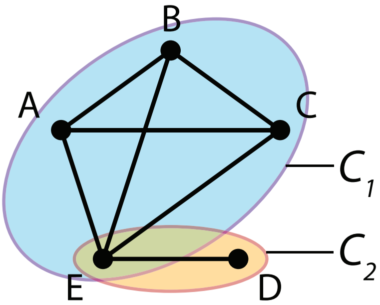
$$\mathbf{S} = \begin{bmatrix} 1.000 & 0.938 & 0.938 & 0.000 & 0.111 \\ & 1.000 & 0.875 & 0.000 & 0.111 \\ & & 1.000 & 0.000 & 0.111 \\ & & & 1.000 & 0.111 \\ & & & & 1.000 \end{bmatrix}$$

Build the matrix $\mathbf{O}$ with $\alpha = 0.10$

$$\mathbf{O} = \begin{bmatrix} 0 & 1 & 1 & 0 & 1 \\ & 0 & 1 & 0 & 1 \\ & & 0 & 0 & 1 \\ & & & 0 & 1 \\ & & & & 0 \end{bmatrix}$$

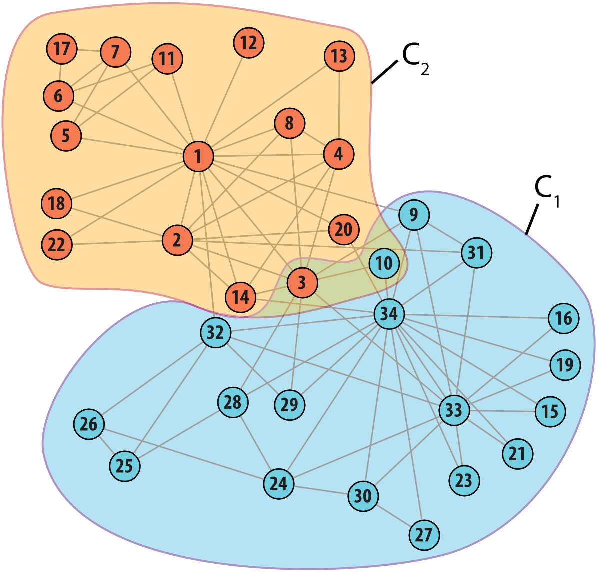


Example: Zachary karate club. Merging communities



$$S_{C_1 C_2} = 0.06 < \alpha = 0.10$$

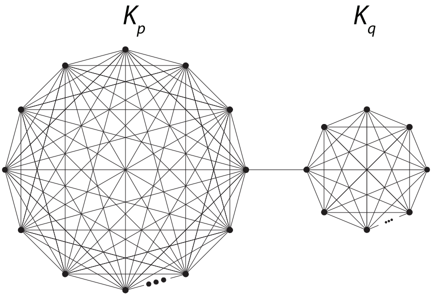
Example: Zachary karate club. Merged communities



Notes on the algorithm[4]

Definition

$G(p,q)$ Consider the network represented by graph $G(p,q)$, with vertex set $V(G(p,q)) = \{1,\dots,p,p+1,\dots,p+q\}$, containing two cliques K_p and K_q, joined by a single edge $(p,p+1)$
$p \geq q \geq 3$

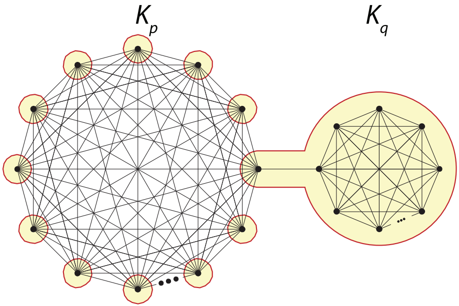


[4] S. Michele Rajtmajer, D. Vukičević, Applied Mathematics and Computation 217 (2010) 3516 - 3521

Theorem

Estrada-Hatano algorithm fails to classify clique K_p as a community in $G(p, q)$ when:

$$\frac{e^q}{(p-2-q)^2} + \frac{1}{(p-2)^2} + \frac{e^{-1.5}}{(p-0.5)^2} - \frac{e^{-1}}{p-1} < 0$$



Amendments to the algorithm.

1. Alternate merging procedure based on a connectivity score:

$$C(A,B) = \frac{|E(A,B)| + 2|A \cap B|}{|A||B|} \tag{6}$$

   - eliminate the arbitrary choice of α;
   - allow the merging of communities that are well connected, but may not overlap;

2. Correcting procedure, intended to test the fitness of obtained communities.

$$M(v,A) = \begin{cases} \frac{1}{|A|-1} \sum_{i \in A} d(v,i) & \text{if } v \in A, \\ \frac{1}{|A|} \sum_{i \in A} d(v,i) & \text{if } v \notin A \end{cases} \tag{7}$$

where $d(v,i)$ the distance from vertex v to $i \in A$

Example: Zachary karate club. Merging communities

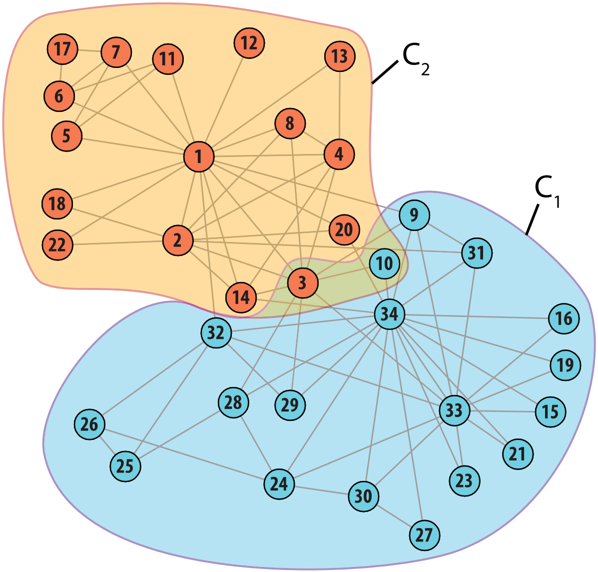
Community-overlap matrix $\mathbf{S}$ for the communities:

$$\mathbf{S} = \begin{bmatrix} 1.000 & 0.938 & 0.938 & 0.000 & 0.111 \\ & 1.000 & 0.875 & 0.000 & 0.111 \\ & & 1.000 & 0.000 & 0.111 \\ & & & 1.000 & 0.111 \\ & & & & 1.000 \end{bmatrix} \qquad \alpha = 0.10$$

Connectivity matrix C for the communities:

$$C = \begin{bmatrix} 0.000 & 0.348 & 0.348 & 0.031 & 0.219 \\ & 0.000 & 0.336 & 0.039 & 0.250 \\ & & 0.000 & 0.027 & 0.219 \\ & & & 0.000 & 0.250 \\ & & & & 0.000 \end{bmatrix} \qquad \overline{C} = 0.207$$

Example: Zachary karate club. Merged communities

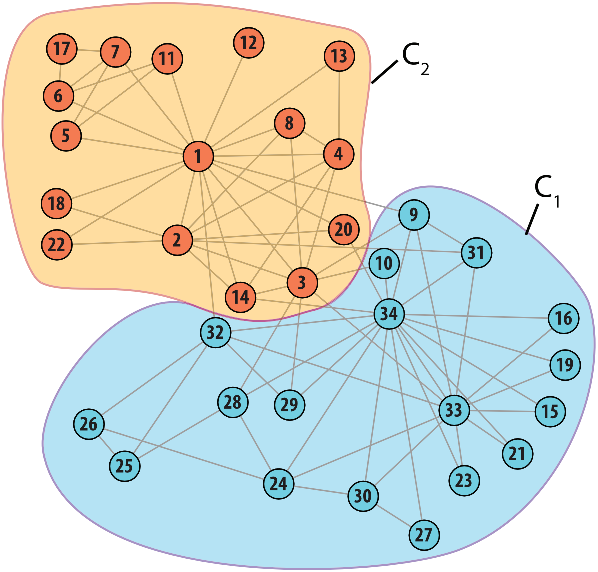


Example: Zachary karate club. Corrective procedure

Consider the case of node 3.

$$
\begin{aligned}
M(v, C_1) &= \frac{1}{|C_1| - 1} \sum_{i \in C_1} d(v, i) = \frac{27}{16} = 1.688 \\
M(v, C_2) &= \frac{1}{|C_2| - 1} \sum_{i \in C_2} d(v, i) = \frac{33}{18} = 1.833
\end{aligned} \tag{8}
$$

Final division after community fitness test



References

E. Estrada, N. Hatano, Communicability graph and community structures in complex networks, Applied Mathematics and Computation 214 (2) (2009) 500 - 511.

E. Estrada, N. Hatano, Communicability in complex networks, Physical Review E 77 (2008) 036111.

C. Bron, J. Kerbosch (1973), Algorithm 457: finding all cliques of an undirected graph, Commun. ACM (ACM) 16 (9): 575 - 577

S. Michele Rajtmajer, D. Vukičević, Applied Mathematics and Computation 217 (2010) 3516 - 3521