Discovering Communities in Social Networks

J.R. Busch, M.G. Beiró, J.I. Alvarez-Hamelin

Facultad de Ingeniería UBA

December 17th, 2009

Complex Networks and Data Communications Group
http://cnet.fi.uba.ar

Topics

1. Community Structure
2. Modularity Optimization
   - The problem
   - The tool: Modularity. Definition
   - Modularity Analysis
3. Submodularity
   - Definitions
   - Weakly optimal and submodular partitions
   - Our algorithm
   - Studying resolution limit
4. Results & Conclusions
   - Numerical results
   - $Q(t)$ vs. t evolution
   - Conclusions

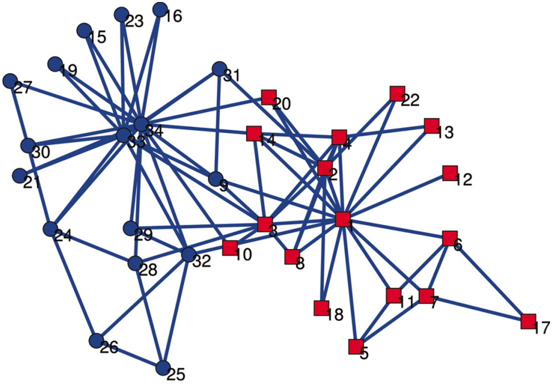
Topics

Community Structure

- Groups of nodes with dense connections among them and few connections with other groups.
- These structures are found in many networks in real life.
- They may help us to understand
  - Systems behavior: Response to an external force
  - Interaction: How members cooperate with each other
  - Group function: Grouping according to functionality

Community Structure
System behavior

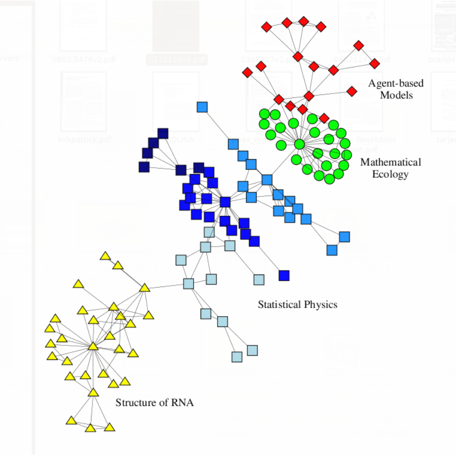
Example: Zachary's Karate Club



Community Structure

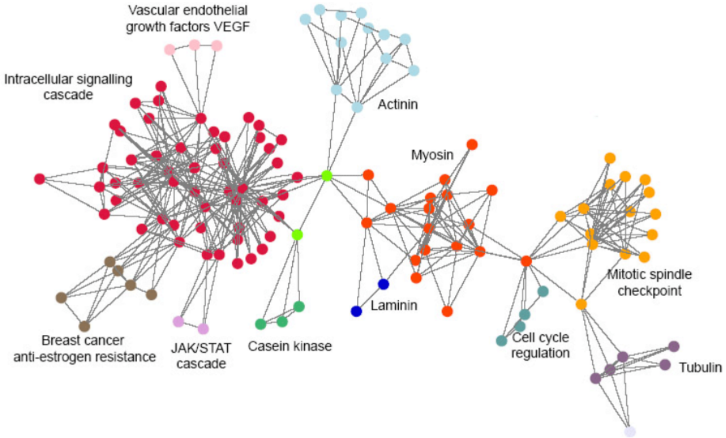
Members interaction

Scientific-Collaboration Network



Community Structure
Group Function

Protein interaction network



Community Structure
Approaches

Formal problem statement:

- Find a “good” graph partition

Several methods:

- Betweenness
- Hierarchical clustering
- Modularity optimization

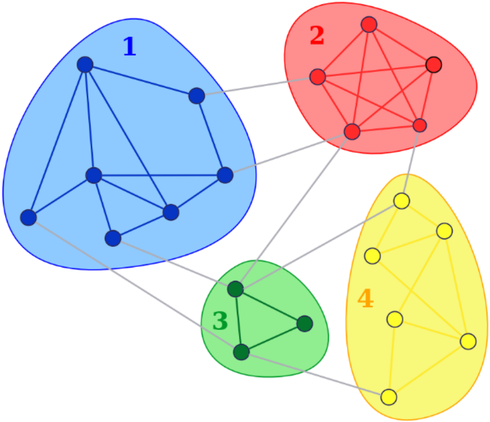
- Modularity has established as a standard measure of quality of a community structure
- Our work:
  - Study its limitations
  - Propose a modularity-based optimization

Topics

Topics

Mathematical Model

Given an undirected graph $G = (V, E)$, find a partition of V, $\mathcal{C} = (C_1, C_2, ..., C_n)$



Topics

Some definitions

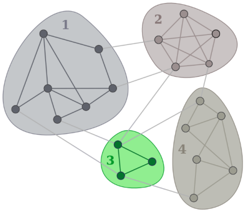
- $k(v)$: Degree of node v
- $m(v, w)$: 1 if (v, w) connected, 0 otherwise
- Remark: if $(v, w) \in E$ then $(w, v) \in E$ too!

We define

$$k(C) = \sum_{v \in C} k(v) = 11 \tag{1}$$

$$n(C) = \sum_{v,w \in C} m(v, w) = 6 \tag{2}$$

$$Q(\mathcal{C}) = \sum_{C \in \mathcal{C}} \left(\frac{n(C)}{k(V)} - \frac{k^2(C)}{k^2(V)} \right) \tag{3}$$



Topics

Modularity analysis

Defining

$$Q(C) = \frac{n(C)}{k(V)} - \frac{k^2(C)}{k^2(V)} \tag{4}$$

First term: fraction of edges internal to communities in G
Second term: the same for a random graph where edges (v, w) were set with probability $p \propto k(v) \cdot k(w)$
Then:

$$Q(\mathcal{C}) = \sum_{C \in \mathcal{C}} Q(C) \tag{5}$$

It can be shown that

$$-\frac{1}{2} \leq Q(\mathcal{C}) \leq 1 \tag{6}$$

Modularity analysis

Given $\mathcal{C}$, we pick an edge $e = (L, R)$ from E randomly.
What's the chance of $R \in C$?

$$P(R \in C) = \frac{k(C)}{k(V)} \tag{7}$$

In the same way...

$$P(L \in C) = \frac{k(C)}{k(V)} \tag{8}$$

And what's the chance of (L, R) *inside* C?

$$P(L \in C \wedge R \in C) = \frac{n(C)}{k(V)} \tag{9}$$

Amazingly:

$$Q(\mathcal{C}) = \sum_{C \in \mathcal{C}} P(L \in C \wedge R \in C) - P(L \in C) \cdot P(R \in C) \tag{10}$$

So we understand modularity as a **sum of covariances** *between L being in each community and R being in the same*

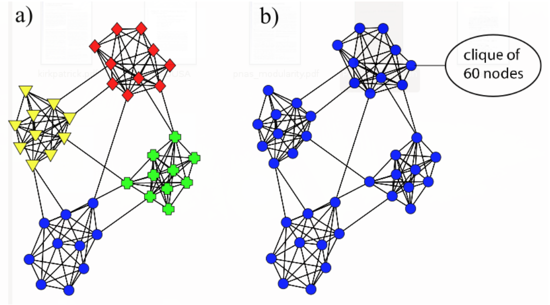
Pros

- Is extensively used in many algorithms and it works
  - As objective function in different optimization approaches
  - As a measure for evaluation and comparison of methods
  - It's easy to recompute on moving over the solution space
- Proved to be fast
- Works fine for many real networks

Cons

- NP-complete (Brandes *et al.*, “On modularity clustering”, 2008)
- Does not arise from a natural community definition
- This gives place to unnatural behaviours sometimes

Resolution limit: Observed by Fortunato & Barthelemy, “Resolution limit in community detection”, 2006



(Kumpula *et al.*, 2007)

Topics

Topics

Submodularity

We define:

$$Q(t, C) = \frac{n(C)}{k(V)} - t \cdot \frac{k^2(C)}{k^2(V)} \tag{11}$$

$$Q(t, C_1, C_2) = \frac{n(C_1, C_2)}{k(V)} - t \cdot \frac{k(C_1) \cdot k(C_2)}{k^2(V)} \tag{12}$$

$$Q(t, C_1 \cup C_2) = Q(t, C_1) + Q(t, C_2) + 2 \cdot Q(t, C_1, C_2) \tag{13}$$

- t is a sort of resolution parameter.
- The second term penalizes big communities, so as t decreases communities will be joined *(zoom out)*

Topics

Weak optimality and submodularity

We say that a partition $\mathcal{C}$ is weakly optimal if joining its sets modularity is not improved.

$$\mathcal{D} \leq \mathcal{C} \to Q(\mathcal{D}) \leq Q(\mathcal{C}) \tag{14}$$

Lemma:

$\mathcal{C}$ is weakly optimal $\iff \forall C_1, C_2 : Q(t, C_1 \cup C_2) \leq Q(t, C_1) + Q(t, C_2)$

Or in other terms:

$$\forall C_1, C_2 : Q(t, C_1, C_2) \leq 0 \tag{15}$$

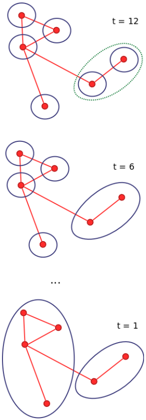
When a partition satisfies this condition we call it SUBMODULAR.

Topics

1. Community Structure
2. Modularity Optimization
   - The problem
   - The tool: Modularity. Definition
   - Modularity Analysis
3. Submodularity
   - Definitions
   - Weakly optimal and submodular partitions
   - Our algorithm
   - Studying resolution limit
4. Results & Conclusions
   - Numerical results
   - $Q(t)$ vs. t evolution
   - Conclusions

A new algorithm for modularity optimization

Building a submodular partition for $t = 1$



For submodular partitions it holds that:

$$\frac{n(C_1, C_2)}{k(V)} - t \cdot \frac{k(C_1) \cdot k(C_2)}{k^2(V)} \leq 0$$

Initial partition: each node alone. (submodular for t big enough)

Big enough: $t = t(\mathcal{C}) = max \frac{n(C_1, C_2) \cdot k(V)}{k(C_1) \cdot k(C_2)}$

do {

Given a submodular partition for a certain t:

Choose some pair $\{C_1, C_2\}$ for which $Q(t(\mathcal{C}), C_1, C_2) = 0$

Now Join: $\mathcal{C}' = \mathcal{C} \setminus \{C_1, C_2\} \cup \{C_1 \cup C_2\}$

$\mathcal{C}'$ is submodular for $t(\mathcal{C})$, so $t(\mathcal{C}') \leq t(\mathcal{C})$

} repeat until $t \leq 1$ or $|\mathcal{C}| = 1$

t is decreasing. Process goes on until $t = 1$ (modularity) or we get a single community.

Topics

Resolution Limit

Studying submodularity we realized that from

$Q(t, C_1, C_2) \leq 0$ (for two communities to remain separate)

it follows that

$$4k(V) \cdot n(C_1, C_2) \leq (k(C_1) + k(C_2))^2$$

So a small subset may stand as a community *if all its connections are to big communities*.

There may be small communities, but they will not be connected.

Topics

1. Community Structure
2. Modularity Optimization
   - The problem
   - The tool: Modularity. Definition
   - Modularity Analysis
3. Submodularity
   - Definitions
   - Weakly optimal and submodular partitions
   - Our algorithm
   - Studying resolution limit
4. Results & Conclusions
   - Numerical results
   - $Q(t)$ vs. t evolution
   - Conclusions

Topics

1. Community Structure
2. Modularity Optimization
   - The problem
   - The tool: Modularity. Definition
   - Modularity Analysis
3. Submodularity
   - Definitions
   - Weakly optimal and submodular partitions
   - Our algorithm
   - Studying resolution limit
4. Results & Conclusions
   - Numerical results
   - $Q(t)$ vs. t evolution
   - Conclusions

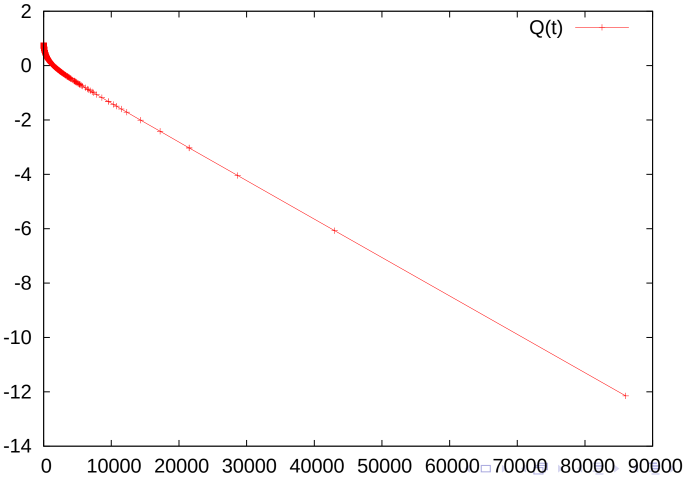
Results for common networks:

Network	Vertices	Edges	NM	BGLL	SM
karate	34	78	0.419	0.419	0.405
jazz	198	2.742	0.442	0.443	0.392
metabolic	453	1.944	-	0.362	0.430
email	1.134	5.145	0.572	0.461	0.521
pgp	10.680	20.287	0.855	0.613	0.865
condmat	15.179	43.011	-	0.865	0.842
cocit	44.968	614.188	-	0.787	0.734
web-bd.edu	325.729	1.103.835	-	0.935	0.960
IR dimes	1.053.396	2.936.840	-	0.851	0.842
web-indochina	7.000.000	195 million	-	0.964	0.979

NM: Newman (quadratic optimization by eigenvalues)
BGLL: Blondel *et al.* (fast unfolding)

Topics

t evolution for *condmat* network.



Topics

- We explained modularity as a *sum of covariances*.
- We proposed a new low-complexity algorithm to find communities based on *submodularity*.
- The algorithm was applied to *big networks* with *good results*.
- From the concept of submodularity we formalized the *resolution limit*.