Semi-average criterion in community detection problems

Shestakov Andrey

Scientific Supervisors:

Boris Mirkin (HSE)

Engelbert Mephu Nguifo (LIMOS)

HIGHER SCHOOL OF ECONOMICS
NATIONAL RESEARCH UNIVERSITY

Outline

1. The task of Community detection
   - Community Structure
   - Methods and algorithms
2. Semi-average clustering criterion
   - Formulation
   - Greedy cluster extraction
   - Search Strategies
3. Comparative coefficients
   - Quality measures
   - Scheme of experiments
   - Results

Main goals

1. Formulate various versions of the community detection algorithm, that optimizes semi-average clustering criterion
2. Develop the platform for comparison experiments

Outline

1. The task of Community detection
   - Community Structure
   - Methods and algorithms
2. Semi-average clustering criterion
   - Formulation
   - Greedy cluster extraction
   - Search Strategies
3. Comparative coefficients
   - Quality measures
   - Scheme of experiments
   - Results

Community detection task

Community Structure

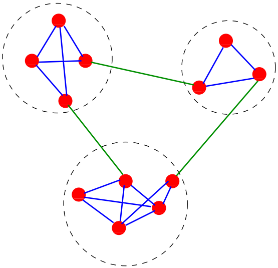


Figure 1 : *Simple example of network with community structure*

- Community structure property is shared be real-world networks as small-world & scale-free properties
- First mentioned in [Girvan and Newman, 2002]

The task of Community detection

- Graph partitioning
  - Cuts
- Hierarchical clustering
  - Agglomerative and divisive approach
- Canonical clustering algorithms
- Spectral clustering
  - Laplace matrix $\mathbf{L} = \mathbf{D} - \mathbf{A}$
- Stochastic algorithms
  - Transition matrix $\mathbf{T} = \mathbf{A}\mathbf{D}^{-1}$
- Modularity optimization
  - $Q = \frac{1}{2m} \sum_{ij} \left(a_{ij} - \frac{k_i k_j}{2m} \right) \delta(\mathcal{C}_i, \mathcal{C}_j)$

Outline

Semi-average clustering criterion

Formulation

Notation

- $\mathbf{A} = \{a_{ij}\}$ – entity-to-entity similarity matrix
- S – cluster (set of points)

Criterion

$$b(S) = \frac{\sum_{\substack{i,j \in S \\ i \neq j}} a_{ij}}{|S|} = (|S| - 1)a(S), \tag{1}$$

$$\text{where } a(S) = \frac{\sum_{\substack{i,j \in S \\ i \neq j}} a_{ij}}{|S|(|S| - 1)} \tag{2}$$

Semi-average clustering criterion

Some derivations

$$\begin{aligned} b(S+k) - b(S) &= \frac{\sum_{i,j \in S \cap k} a_{ij}}{|S|+1} - \frac{\sum_{i,j \in S} a_{ij}}{|S|} = \cdots = \\ &= \frac{2|S|a(k,S) - a(S)(|S|-1)}{|S|+1}, \end{aligned} \tag{3}$$

$$\begin{aligned} b(S-k) - b(S) &= \frac{\sum_{i,j \in S/k} a_{ij}}{|S|-1} - \frac{\sum_{i,j \in S} a_{ij}}{|S|} = \cdots = \\ &= a(S) - 2a(k,S). \end{aligned} \tag{4}$$

$$\text{where } a(k,S) = \begin{cases} \sum_{i \in S} a_{ik} \Big/ (|S|-1) & \text{if } k \in S \\ \sum_{i \in S} a_{ik} \Big/ |S| & \text{otherwise} \end{cases} \tag{5}$$

Semi-average clustering criterion

Formulation

General derivation

$$\Delta_k b(S) = z_k \left[\frac{(|S| + z_k)a(S) - 2\left(|S| + \frac{z_k+1}{2}\right) a(k,S)}{|S| + 1} \right] \quad (6)$$

Semi-average clustering criterion

Properties

- Cluster S is optimal by (1) if $\forall k \in S\ \alpha(k,S) = a(k,S) - \frac{a(S)}{2} > 0$ (determined from (3)-(4))

Similarity matrix adjustments

1. By subtraction of a constant "noise" level π: $\mathbf{A} - \pi$. Usually π is calculated as a mean value over all entities of matrix $\mathbf{A}$
2. By subtracting random interactions. This approach has many in common with Newman's modularity concept: $\mathbf{A}' = \{a_{ij} - k_i k_j / 2m\}$.

Algorithm 1 AddRemAdd(i) algorithm

Input: Adjacency matrix $\mathbf{A} = (a_{ij})$, initial vertex index i
Output: Sub-optimal cluster S

Step 1: Initialization

State $n = |V|$

1: Set n-dimensional vector $\mathbf{z}$ with $z_i = 1$ and $z_j = -1, j \neq i$
2: Find i^* s.t. $a_{ii^*} = \max_j a_{ij}$, set $z_{i^*} = 1$, $n_S = 2$ - cluster cardinality
3: Set $ma = a_{ii^*}$ - the average within-cluster similarity (2)
4: Set $a(i) = a(i^*) = a_{ii^*}$ and $a(j, S) = (a_{ij} + a_{i^*j})/2$ - average similarities of entities to cluster

repeat

  for $v_k \in V$ **do**

    $$d_k = z_k \left[\frac{(n_S + z_k) \cdot ma - 2\left(n_S + \frac{z_k+1}{2}\right) a(k)}{n_S + 1} \right]$$

    Find k^* s.t. $d_{k^*} = \max_k d_k$

Step 3: Update

  if $d_{k^*} > 0$ **then**

    Update $ma = ma + \frac{2z_{k^*}}{n_S - \frac{3z_{k^*}+1}{2}} \left[ma - a(k^*)\right]$

    Update $a(k) = a(k) + z_{k^*} \frac{1}{|S| - \frac{z_k+1}{2} - z_{k^*}} \left[a(k) - a_{kk^*}\right]$ for each $k \neq k^*$

    $n_S = n_S - z_{k^*}$

    $z_{k^*} = -z_{k^*}$

until any $d_k > 0$

Output cluster $S = \{i : z_i = 1\}$ with corresponding average similarity $a(S)$ and criterion value $b(S)$

Search Strategies

Cluster search

Incremental Apply AddRemAdd(i) to all vertices v_i, choose cluster S^* with maximum value of $b(S^*)$

Randomized Choose initial randomly i only once and take the output of AddRemAdd(i)

Community search

Overlapping Additive clusters After obtaining cluster choosing a cluster S, matrix $\mathbf{A}$ is updates as $\mathbf{A} = \mathbf{A} - a(S) z_S z_S^\mathsf{T}$

Non-overlapping clusters After obtaining cluster S, one just remove rows and columns, correspondent to vertices in S from matrix $\mathbf{A}$

The note on unweighted networks

Bad decision making

Possible solutions

- Initialization from dense subset of vertices (n-clique, k-core)
- Recalculation of similarity matrix
  1. Ratio of common neighbours

$$\omega_{ij} = \frac{|N(v_i) \cap N(v_j)|}{|N(v_i) \cup N(v_j)|},$$

  2. Pirson's correlation

$$r_{ij} = \frac{\sum_k (a_{ik} - \mu_i) \sum_k (a_{jk} - \mu_j)}{n \sigma_i \sigma_j}.$$

Outline

Quality measures

Ratio of correctly clustered vertices

$$\varphi^{\text{CCV}} = \sum_{i=1}^{k_a} q_i \Big/ n, \qquad (7)$$

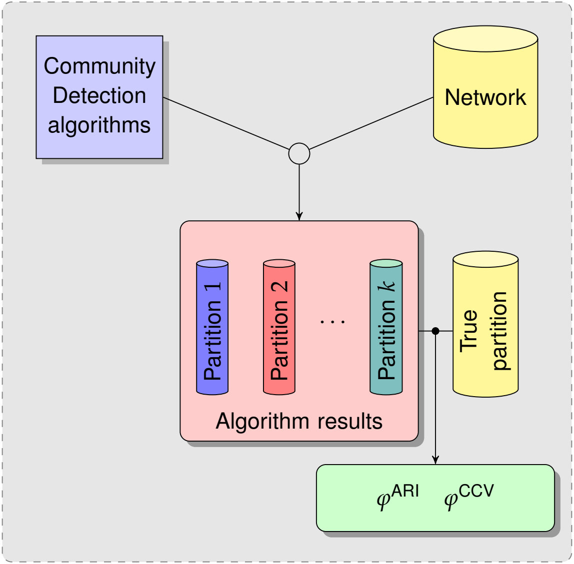
where q_i is the number of correctly clustered vertices of cluster $\mathcal{C}_i$.

Adjusted Rand Index

$$Rand(X, Y) = \frac{a + d}{a + b + c + d} \to \varphi^{\text{ARI}}, \qquad (8)$$

- a # if pairs of vertices, joined by community both in X and Y
- b (c) # if pairs of vertices, found in the same community in X (Y) but in different in Y (X)
- d # if pairs of vertices, found in different communities in both partitions

Scheme of experiments



Algorithms

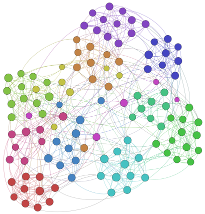
- `EgdBtws` – Girvan and Newman EdgeBetweenness [2004]
- `FastGreedy` – Girvan and Newman greedy Q optimization [2004]
- `LeadEigen` – Newman spectral Q optimization [2006]

Using implementation in `igraph` library for `python`

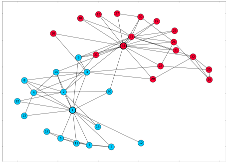
Results

Real networks

- Zackhary's Karate Club
- American Football League
- Dolphin's network



(a) *Football*



(b) *Zackhary's karate club*

Figure 2 : *Examples of real networks with known community structure*

Zackhary's Karate Club

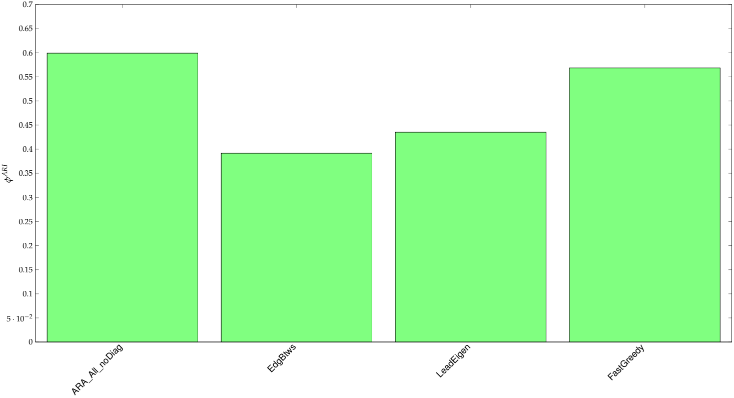


Figure 3 : *ARI index of obtained partitions*

Zackhary's Karate Club

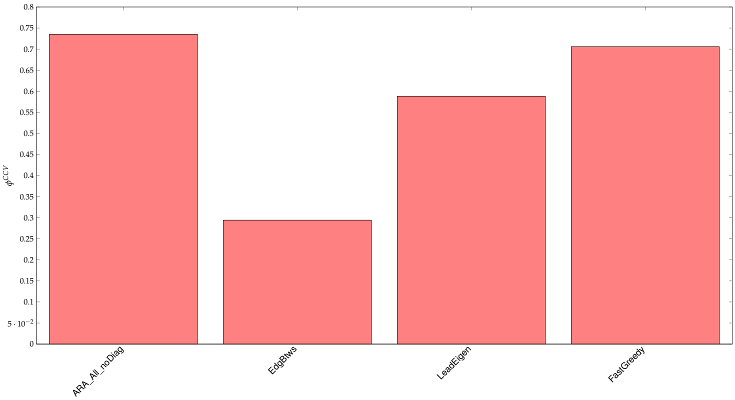


Figure 4 : *CCV of obtained partitions*

American Football League

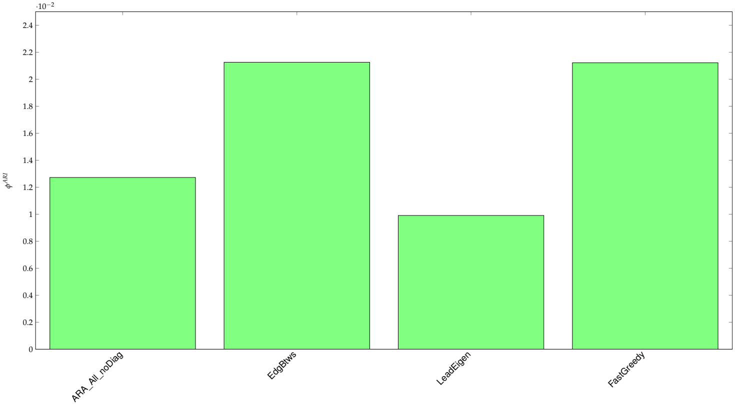


Figure 5 : *ARI index of obtained partitions*

American Football League

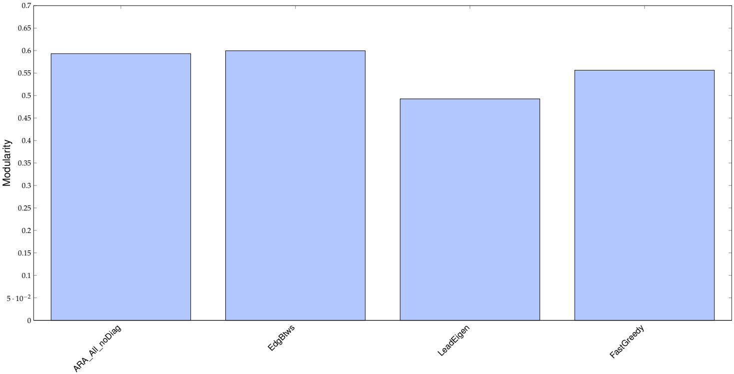


Figure 6 : *Modularity of obtained partitions*

Results

Generated networks

- Proposed in [Lancichinetti and Fortunato, 2009]
- Directed/Undirected, Weighted/Unweighted networks
- Many parameters
  - Topological mixing parameter $k_i^{(\text{in})} = (1 - \mu_t)k_i$

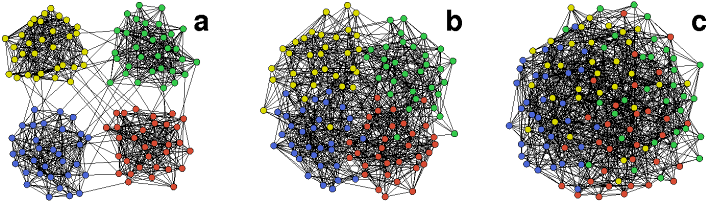


Figure 7 : *4-planted partition model for some* μ_t

Parameter initialization

Standard parameters:

- Number of vertices – 128
- Number of communities – 4
- Size of communities – 32
- Average vertex degree – 16
- Topological mixing μ_t – $[0.1 - 0.7]$

Results

Generated networks

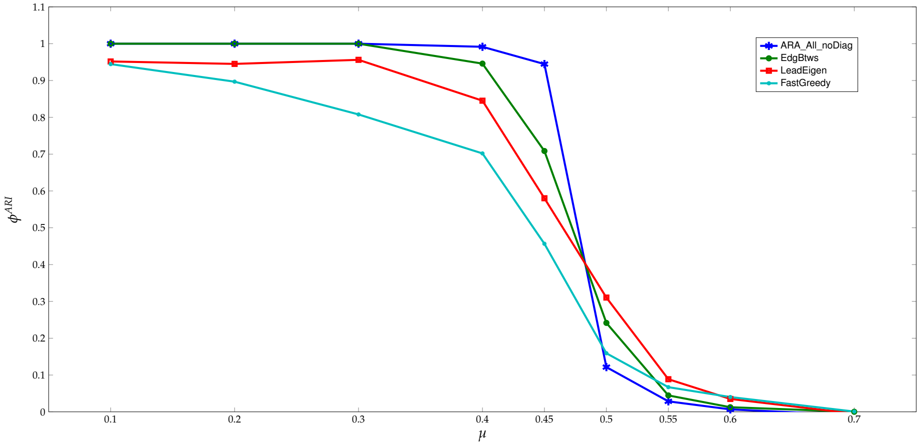


Figure 8 : *Average ARI with change of μ*

Results

Generated networks

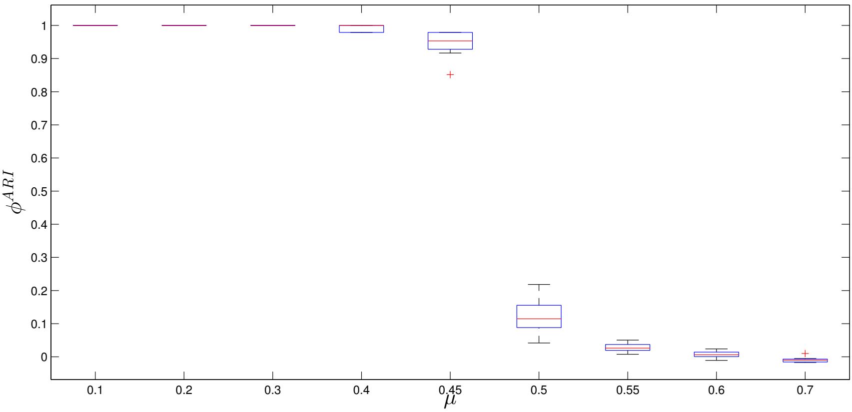


Figure 9 : *Stability of obtained partitions* μ

Conclusion

Main issues

- Can we speed-up?
- Apply on BIG networks?

That‘s all, folks!