Community Detection Algorithms

Leonid E. Zhukov

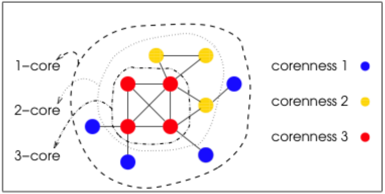
School of Applied Mathematics and Information Science
National Research University Higher School of Economics

14.03.2013

НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ
УНИВЕРСИТЕТ

Finding k-cores

- Set S is a k-core if for $\forall v \in S$, $k(v) \geq k$ and is maximal subgraph.
- The core number of a vertex is the highest order of a core that contains this vertex



Algorithm: Core decomposition

Input: graph G(V,E)

Output: $core[v]$ - core number for each vertex

```
compute deg[v];
sort V : deg[v_{i+1}] ≥ deg[v_i];
for each v ∈ V do
    core[v] = degree[v];
    for each u ∈ NN(v) do
        if deg[u] > deg[v] then
            deg[u] = deg[u] − 1;
            sort V
        end
    end
end
```

Community detection

Definition

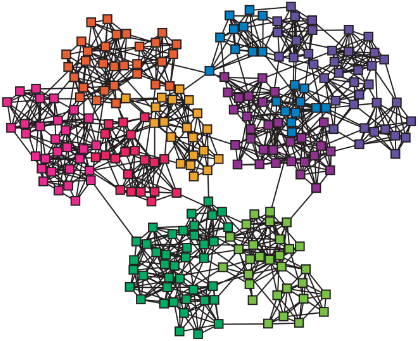
Network community is a group of verticies such that vertices inside the group are connected with many more edges than between groups.

- Consider only sparse graphs $m \ll n^2$
- Each community should be connected
- Divisive algorithms (vs agglomerative)
- Finding communities through graph partitioning
- Graph partition - division of graph vertices into groups such that each vertex belongs only to one group
- Graph cut is a set edges whose end points are in different partitions
- NP-hard, solved by approximation algoritms and heuristicss
- Recursive 2-way partition, multiway partiton

Edge betweenness

Edge betweenness -number of shortest paths $\sigma_{st}(e)$ going through edge e

$$C_B(e) = \sum_{s \neq t} \frac{\sigma_{st}(e)}{\sigma_{st}}$$



Edge betweenness

Newman-Girvan, 2004

Algorithm: Edge Betweenness

Input: graph G(V,E)

Output: Dendrogram

repeat

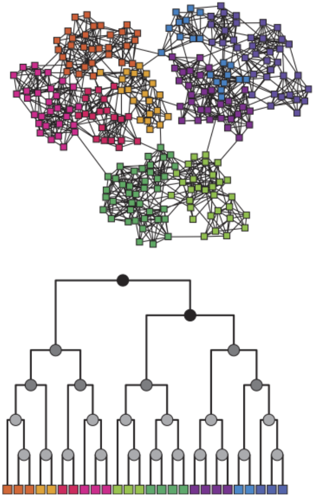
  For all $e \in E$ compute edge betweenness $C_B(e)$;

  remove edge e_i with largest $C_B(e_i)$;

until *edges left* ;

If bi-partition, then stop when graph splits in two components (check for connectedness)

Dendrogram



Modularity

- Modularity:

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j)$$

  c_i- class, $\delta(c_i, c_j)$- kronecker delta

- Single class, $\delta(c_i, c_j) = 1$

  $Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) = 0$

-

Modularity

- e_{rs} - fraction of edges between vertices of types r and s (edges between groups):

$$e_{rs} = \frac{1}{2m} \sum_{i \in R, j \in S} A_{ij} = \frac{1}{2m} \sum_{ij} A_{ij} \delta(c_i, r) \delta(c_j, s)$$

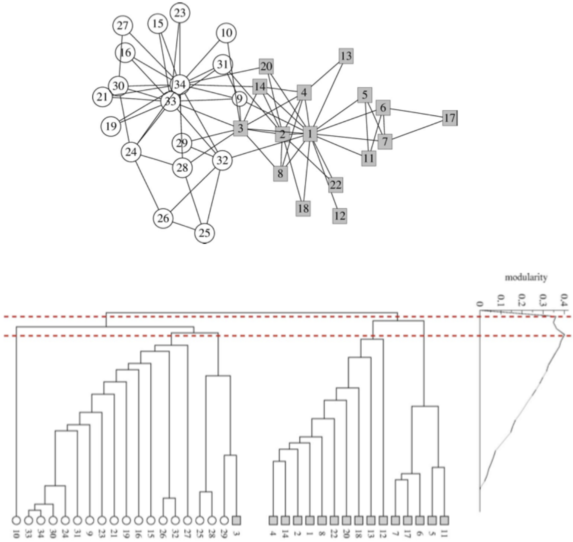
- a_r - fraction of edges attached to vertices of type r (edges to all vertices):

$$a_r = \frac{1}{2m} \sum_{i \in R, j} A_{i,j} = \frac{1}{2m} \sum_{i} k_i \delta(c_i, r)$$

- Modularity

$$Q = \frac{1}{2m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) \delta(c_i, c_j) = \sum_r (e_{rr} - a_r^2)$$

Dendrogram and modularity score



Spectral Modularity Maximization

- Direct modularity maximization (bi-partitioning), [Newman, 2006]
- For two classes C_1, C_2 indicator variable $s = \pm 1$

$$\delta(c_i, c_j) = \frac{1}{2}(s_i s_j + 1) = \begin{cases} 1: & i, j \in C_1 \text{ or } i, j \in C_2 \\ 0: & i \in C_1, j \in C_2 \text{ or } i \in C_2, j \in C_1 \end{cases}$$

- Modularity

$$Q = \frac{1}{4m} \sum_{ij} \left(A_{ij} - \frac{k_i k_j}{2m} \right) (s_i s_j + 1) = \frac{1}{4m} \sum_{i,j} B_{ij} s_i s_j$$

where

$$B_{ij} = A_{ij} - \frac{k_i k_j}{2m}$$

Spectral Modularty Maximization

- Qudratic form:

$$Q(\mathbf{s}) = \frac{1}{4m}\mathbf{s}^T\mathbf{B}\mathbf{s}$$

- Integer optimization - NP, relaxation $s \to x$, $x \in R$
- Keep norm $||x||^2 = \sum_i x_i^2 = \mathbf{x}^T\mathbf{x} = n$
- Quadratic optimization

$$Q'(\mathbf{x}) = \frac{1}{4m}\mathbf{x}^T\mathbf{B}\mathbf{x} - \lambda(\mathbf{x}^T\mathbf{x} - n)$$

- Eigenvector problem

$$\mathbf{B}\mathbf{x}_i = \lambda'_i\mathbf{x}_i$$

- Approximate modularity

$$Q'(\mathbf{x_i}) = \frac{n}{4m}\lambda_i$$

- Maximization - maximal λ

Spectral Modularty Maximization

- Can't choose $\mathbf{s} = \mathbf{x}$k, can select optimal $\mathbf{s}$
- Decompose in the basis: $\mathbf{s} = \sum_j a_j \mathbf{x_j}$, where $a_j = \mathbf{x}_j^T \mathbf{s}$
- Modularity

$$Q(\mathbf{s}) = \frac{1}{4m} \mathbf{s}^T \mathbf{B} \mathbf{s} = \frac{1}{4m} \sum_i (\mathbf{x}_i^T \mathbf{s})^2 \lambda_i$$

- $\max Q(\mathbf{s})$ reached when $\lambda_1 = \lambda_{max}$ and max $\mathbf{x}_1^T \mathbf{s} = \sum_j x_{1j} s_j$
- Choose $\mathbf{s} \parallel \mathbf{x}_1$, $\mathbf{s} = sign(\mathbf{x}_1)$

Modularity maximization

Algorithm: Spectral moduldarity maximization: two-way parition

Input: adjacency matrix $\mathbf{A}$

Output: class indicator vector $\mathbf{s}$

compute $\mathbf{k} = deg(\mathbf{A})$;

compute $\mathbf{B} = \mathbf{A} - \frac{1}{2m}\mathbf{k}\mathbf{k}^T$;

solve for maximal eigenvector $\mathbf{B}\mathbf{x} = \lambda\mathbf{x}$;

set $\mathbf{s} = sign(\mathbf{x}_1)$

Recursive bisection

References

- An O(m) Algorithms for Cores Decomposition of Networks. V. Batagelj, M. Zaversnik, 2002
- Finding and evaluating community structure in networks, M.E.J. Newman, M. Girvan, Phys. Rev E, 69, 2004
- Modularity and community strcuture in networks, M.E.J. Newman, PNAS, vol 103, no 26, pp 8577-8582, 2006